

TECHNISCHE UNIVERSITEIT EINDHOVEN
Department of Mathematics and Computer Science

MASTER'S THESIS

ANONYMOUS AND FUZZY
IDENTITY-BASED ENCRYPTION

by
P.P. van Liesdonk

Supervisors:
L.A.M. Schoenmakers
P. Tuyls

Eindhoven, Augustus 2007

Acknowledgements

This master thesis is the result of the 9 months final internship of Peter van Liesdonk at Philips Research at the High Tech Campus in Eindhoven, at the department ‘Information & System Security’. The topic of this report is the study of identity-based encryption with fuzzy keys, with a focus on the anonymity of ciphertexts.

First, I would like to thank my supervisor at Philips Research, Pim Tuyls, and my supervisor at the Technische Universiteit Eindhoven, Berry Schoenmakers for their guidance. I would also like to thank the ISS department for giving me the opportunity to do an internship, and all the employees and interns at ISS for giving me a very enjoyable environment to work in.

I also want to thank my family for their support and for listening to me, even when they had no idea what I was talking about.

Finally I want to thank Susanne Dams and my roommates for giving me some necessary distractment from my work.

Contents

Acknowledgements	1
Table of Contents	4
1 Introduction	5
1.1 Anonymous Fuzzy Identity-Based Encryption	5
1.2 Background	6
1.3 Our contributions	8
1.4 Organization	8
2 Preliminaries	11
2.1 Notation	11
2.2 Pairings	12
2.3 Bilinear Diffie-Hellman	13
2.4 Relaxing Hashing Requirements	14
3 Security Notions	15
3.1 Security Goals	15
3.2 Attack Models	18
3.3 Random Oracle Model	20
4 Identity-Based Encryption	23
4.1 Applications	24
4.2 Boneh-Franklin’s <code>BasicIdent</code> scheme	25
5 Fujisaki-Okamoto Transformation	27
5.1 A Weak Game	28
5.2 Security Under Chosen Plaintext Attacks	29
5.3 Chosen Ciphertext Attacks	33
6 Fuzzy Identity-Based Encryption	37
6.1 Introduction	37
6.2 Basic Scheme	38
6.3 Chosen Plaintext Indistinguishability	39
6.4 Chosen Plaintext Anonymity	42
6.5 Chosen Ciphertext Attacks	43

7	Conclusions	47
7.1	Comparison with Sahai-Waters	47
7.2	Further Research	48
	Bibliography	53
A	Alternative Fuzzy Scheme	55

Chapter 1

Introduction

1.1 Anonymous Fuzzy Identity-Based Encryption

In this thesis we will introduce a new concept for public key encryption, called *anonymous fuzzy identity-based encryption*. In this form of encryption, a descriptive set of attributes is used to encrypt a message. Decryption is performed using a secret key that corresponds to the set of attributes.

In contrast to regular public key encryptions, we want to allow a certain tolerance in the key. That means that when the set of attributes used for encryption does not completely match the set of attributes that correspond to the secret key, decryption is still possible. However when the match is lower than a certain threshold, decryption should not be possible anymore. This is the *fuzzy* part.

As second property we desire is the following: if someone intercepts the encrypted message, he should not be able to learn any information about the set of attributes used to do the encryption. This leads to truly *anonymous* communication, for example with the use of a public bulletin board.

A scheme like this brings about several interesting applications:

Using biometrics in Identity-Based Encryption The first is an identity-based encryption scheme based on biometric information. That is we can view a user's biometric, for example an iris scan, as that user's identity described by several attributes and then encrypt to the user using their biometric identity. Biometric measurements are noisy, but the error-tolerance property of the Fuzzy IBE allows for a secret key (derived from a measurement of a biometric) to decrypt a ciphertext encrypted with a slightly different measurement of the same biometric. In many situations, this form of IBE has a number of important advantages over 'standard' IBE.

The process of obtaining a secret key from an authority is very natural and straightforward. In standard IBE, a user with a certain identity (for example 'bob@domain.com') will need to go to an authority to obtain the private key corresponding to the identity. In this process he will need to prove in some way that he is indeed entitled to this identity. This type of authentication is not always clear and the robustness of this process is questionable. In contrast, when using a biometric the user must demonstrate ownership of his biometric under supervision of an operator. This process is clear and simple and the robustness is limited only to the quality of the biometric used.

Also, a biometric is an inherent trait and will always be with a person. Using biometric

will mean that the person will always have their public key with them. This is convenient in some situations where a person wants to present an encryption key to someone when they are physically present. The recipient of the public key immediately knows that he did not receive someone else's public key and that any encryption he would make can only be read by the person he physically met. Afterwards, the two can exchange messages completely anonymous.

Finally, using a biometric as identity assures that the identities are unique if the underlying biometric is of good quality. Clearly this is not the case for some standard identities like 'Bob Smith'.

Attribute-based encryption In this application a party will wish to encrypt a document to all users that have a certain set of attributes. For example, in a company, a manager wants to encrypt a document so it is accessible to everybody in the management department and additionally to people who have special clearance.

He now encrypts his message with the attributes 'management' and 'clearance=yes'. Every employee who has a secret key containing one of these attributes is able to decrypt the message. However, employees with neither of these attributes cannot decrypt the message, and are not even able to distinguish who is able to read it.

The mayor difference with just making multiple encryptions is that it is possible to create all sorts of access schemes. Additionally, individual attributes can be revoked without knowledge of the sender.

Public key encryption with fuzzy keyword search Suppose Bob sends an encrypted email to Alice. This encrypted mail ends up on Alice's email gateway. However, Alice receives lots of mail, and she wants to receive only mail that has been flagged as 'urgent' directly on her phone.

In this application Alice gives a small message containing 'urgent' to her mail gateway; the gateway is then able to scan all her mail to find only the ones that match and forward those. This is an application first introduced by [BCOP04]

The interesting part of an anonymous scheme is that the mail gateway does not know which keyword Alice wanted it to search for. In a fuzzy scheme, Alice can also search for combinations of multiple keywords, or for keywords that have spelling mistakes.

In this thesis we introduce an encryption scheme that works for the first application. In Appendix A we give some insights in how the other applications could be made possible.

1.2 Background

Identity-based encryption (IBE) A common feature of most public key encryption settings is that both the public and the secret key need to be generated. If someone wants to send an encrypted e-mail then he first needs to obtain this users public key in some way. Then even if he has obtained this person's public key he has no direct way of knowing whether it is the correct key, since there is no direct connection between this person and the key. There exist certificates and Public-Key Infrastructures to solve this last problem, but a person still has to setup his keys, get certificates from a PKI and distribute the certificate before he can receive encrypted e-mails.

In 1984 Shamir [Sha84] asked for a public key encryption scheme in which the public key can be an arbitrary string. Shamir's original motivation for this encryption scheme was to simplify this certificate management. For example, if you want to send an encrypted mail to a

recipient, e.g. bob@domain.com, the encryption can be done without requiring the existence of a PKI or other way to obtain Bob's public key.

It was much later that Boneh and Franklin [BF01] presented the first identity-based encryption scheme that was both practical and secure. Their solution featured a novel use of efficiently computable bilinear maps and was provably secure in the random oracle model (see Section 3.3.)

Soon after, Canetti et al. [CHK03] proposed a construction for IBE that was provable outside the random oracle model, followed by more efficient schemes by Boneh and Boyen [BB04]. These were however all proven secure in a weaker model known as the Selective-ID model.

Anonymous encryption In 2001, Bellare et al. [BBDP01] treats the notion of *anonymity* or *key privacy* for a cryptosystem. In this notion the ciphertext of an encryption may not give any information on the key that was used to perform the encryption, in addition to the privacy of the message.

This might sound odd, especially since in a public key setting where the encryption key is public: there might seem nothing to keep secret. However, the privacy refers to the information conveyed to the adversary regarding which specific key - out of a set of known public keys - is the one under which a given ciphertext was created. This is called anonymity because it means that the receiver is anonymous from the point of view of the adversary.

A scheme that has the property of anonymity has several uses. The most obvious one is truly anonymous communication, for example by exchanging encrypted messages via a public bulletin board. An adversary observing the bulletin board can only see the ciphertext. Aside from not being able to find information on the plaintext or the origin of the message, he now also has no information on the intended recipient.

Combining anonymity with identity-based encryption is a logical step. The Boneh-Franklin scheme [BF01] is already anonymous, though it was only in 2006 that Boyen and Waters proposed a scheme that was provably secure in the standard model [BW06].

This has a direct application in 'searchable encryption' [SWP00]. In 2004, Boneh et al. [BCOP04] introduced their form of searchable encryption, namely public key encryption with keyword search (PEKS). Their solution is based on an anonymous identity-based encryption scheme. Later [ABC⁺05] constructed a general framework to create PEKS from anonymous identity-based encryption schemes.

Fuzzy identity-based encryption There has been lots of work in applying biometric to cryptography. This is mainly focused on the derivation of a secret from a biometric. [DRS04, APM04]. In this setting there is an explicit assumption that the biometric must remain a secret.

One can argue whether this assumption is realistic. After all, biometric information can be stolen or forged. Fingerprints, for example, are left everywhere and can be easily lifted. But since biometrics can identify a person uniquely, it makes sense to use them as the public key in an identity-based encryption scheme.

The problem with this approach is that biometrics usually consist of noisy data, i.e. two measures ω and ω' of the same biometric are not completely the same. Usually the best one can do is state that are within a certain distance of each other. This is in contrast to regular IBE schemes, which view

Sahai and Waters [Wat05] took these ideas to formalize the concept of *fuzzy identity-based encryption* and make an encryption scheme. They proved this scheme to be selective-ID secure

(see Section 3.2.) A disadvantage is that the ciphertext necessarily included the complete public key used for encryption, resulting in a scheme that is not anonymous.

1.3 Our contributions

In this thesis, we took the concepts from [Wat05] and try to make a scheme that is also anonymous. However, their specific approach can not be generalized to an anonymous setting.

Therefore we go to a different approach for a fuzzy scheme, with several parallel IBE schemes. [Wat05] discussed this approach and rejected the idea for being too vulnerable to collusion attacks. Yet we thought it was worth a second look. By looking at a weaker setting we were able to construct an anonymous fuzzy IBE.

This weaker setting indicates that instead of viewing an identity as a set of attributes, we view an identity as a combination of a name and a set of attributes. In this way we can prevent the mentioned collusions attacks. However, it also limits the applications of the scheme. It makes the second and third application from the introduction impossible, and severely hinders the first.

Luckily here are strong indications that with newer IBE techniques like probabilistic secret key generation [BW06] this approach can still be generalized.

Our second contribution is the use of Fujisaki-Okamoto [FO99]. This transformation is often used to create a highly secure encryption scheme from a much weaker one. However, this is often done in ways that are very specific to one scheme (e.g. [BF01]) or without giving a proof or arguments for feasibility (e.g. [Wat05]).

In Chapter 5, we give an extensive proof that the transformation from [FO99] can be used in the construction of both anonymous and secure identity based encryptions schemes.

1.4 Organization

This thesis is organized as follows:

In Chapter 2, we start with discussing some preliminaries. First we explain the notation used throughout this thesis. We then introduce the concept of pairings and give the definition of the Bilinear Diffie-Hellman assumption. Finally, we explain how to use cryptographic hash functions in conjunction with pairings.

Chapter 3 gives an overview of the security notions we will use throughout this paper. We do this using a separation between security goals and attack models, as proposed by Naor. We introduce a new security goal that we will use in Chapter 5. We end the chapter with an explanation of the random oracle model and a discussion of its use.

Chapter 4 introduces the concept of identity-based encryption. We give an overview of its applications and formalize the concept of identity-based encryption. We then discuss the Boneh-Franklin cryptosystem and some of its properties.

In Chapter 5 we explain the transformation that Fujisaki and Okamoto proposed to make strong cryptosystems out of weak ones. We give a new proof to show that the transformation can be applied to an identity-based setting and that it can be used to create anonymous schemes.

Chapter 6 introduces the concept of fuzzy identity-based encryption. We start with a formalization of the setting. We then introduce a new encryption scheme that implements

this concept and prove it is secure in a weak model. The Fujisaki-Okamoto transformation will then be used to create a much stronger, provably secure, cryptosystem.

Finally, Chapter 7 contains a review of the work in this thesis. We give a recap of the contributions we made and discuss the problems that are still open.

In addition, we introduce another fuzzy identity-based scheme in Appendix A, which is more general than the one in Chapter 6. However, we give no complete proof of security; only some arguments to make the security plausible.

Chapter 2

Preliminaries

In this chapter we introduce some notions used throughout the report. We begin with formalizing the notation we use for the description of encryption schemes and arithmetic operations.

We proceed with a brief review of pairings and their properties and discuss the computational problem called the *bilinear Diffie-Hellman assumption* on which the security of our schemes is based.

2.1 Notation

We use the following notations throughout this paper:

With \mathbb{G} we denote a group, where 1 is the identity element and $\log_g(A)$ is the discrete logarithm of A to base generator g . We let \mathbb{G}^* denote the set of non-identity elements of \mathbb{G} . For reading clarity, we choose to denote all group operations in multiplicative form; even operations on elliptic curves, which are usually written additive. The only exception here are bitstrings, for which we use the *exclusive or* (\oplus) operation.

We denote an asymmetric encryption scheme as $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$. Here

- $\mathcal{G}(\kappa)$ is the *Setup* algorithm, which takes security parameter κ to generate all parameters required to use the scheme; most of these are public, but sometimes it also generates master secrets.
- $\mathcal{K}()$ is the *Key-generation* algorithm, which generates a public / secret key pair (pk, sk) ; in identity-based encryption it takes an identity id as parameter, otherwise it takes no parameters.
- $\mathcal{E}_{pk}(m; r)$ is the *Encryption* algorithm, which takes a plaintext m and returns a ciphertext under public key pk , where r is a random nonce.
- $\mathcal{D}_{sk}(c)$ is the *Decryption* algorithm, which takes a ciphertext c and returns the plaintext using secret key sk .

We denote a symmetric encryption scheme by E^{sym} . A specific encryption of a message m under symmetric key k is written as $E_k^{sym}(m)$. A decryption of a ciphertext c is written as $D_k^{sym}(c)$.

If anything in these notations is irrelevant or inherently clear, it may be omitted.

2.2 Pairings

A pairing is a function that takes as input two points on an elliptic curve and outputs an element of some multiplicative group, also satisfying some special properties. Due to these properties, pairings are hard to construct. In 1993, Menezes et al. [MOV93] discovered that the Weil pairing can be used to attack discrete logarithm-based systems on a certain class of elliptic curves; the so-called MOV-reduction. One year later, Frey and Rück used the Tate pairing [FMR99] to describe a similar attack. In 2000 Joux [Jou00] discovered that pairings can be used as cryptographic building blocks as well.

Formally, let \mathbb{G}_1 , \mathbb{G}_2 and \mathbb{G}_T be groups of prime order p . Since the order of these groups is prime, they are all cyclic. The last group is called the target group, hence the subscript. In practice \mathbb{G}_1 and \mathbb{G}_2 are points on an elliptic curve and \mathbb{G}_T is a scalar field.

Definition 2.1. A *pairing* is an efficiently computable map $e : \mathbb{G}_1 \times \mathbb{G}_2 \mapsto \mathbb{G}_T$ satisfying the following conditions:

Bilinear: for every $g_1, h_1 \in \mathbb{G}_1$, $h_2 \in \mathbb{G}_2$

$$e(g_1 h_1, g_2) = e(g_1, g_2) e(h_1, g_2),$$

and for every $g_1 \in \mathbb{G}_1$, $g_2, h_2 \in \mathbb{G}_2$

$$e(g_1, g_2 h_2) = e(g_1, g_2) e(g_1, h_2).$$

Non-degenerate: There exist $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ such that

$$e(g_1, g_2) \neq 1.$$

We can find groups for which these properties hold: the modified Weil pairing [BF01] and the Tate pairing prove the existence of the existence of groups (see [Jou02] for a survey on which curves can be used.) Due to an algorithm by Miller [Mil86], they can be computed efficiently. They are, however, mathematically complex and outside the scope of this report. Fortunately, the abstract definition satisfies to create quite powerful schemes.

It is possible that $\mathbb{G}_1 = \mathbb{G}_2$. This is called the symmetric case, while the asymmetric case is the more general one in which \mathbb{G}_1 and \mathbb{G}_2 are possibly different. The asymmetric case allows the use of different groups which can bring some performance benefits. There are no known pairings with $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}_T$.

We can fix an isomorphism $\psi : \mathbb{G}_2 \mapsto \mathbb{G}_1$. In the symmetric case, this is the identity map. Note that if g_2 is a generator of \mathbb{G}_2 , then $\psi(g_2)$ is a generator of \mathbb{G}_1 .

The following is the central algebraic property of pairings that is responsible for a lot of their power.

Proposition 2.2. For any $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and any $a_1, a_2 \in \mathbb{Z}_p$

$$e(g_1^{a_1}, g_2^{a_2}) = e(g_1, g_2)^{a_1 a_2}.$$

Proof. This follows easily from the bilinearity property. □

2.3 Bilinear Diffie-Hellman

From now on, we will work with symmetric pairings $e : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_T$. This gives us the possibility to define the Bilinear Diffie-Hellman problem (BDH), on which we will base the security of our schemes. We must note that this has as an implication to efficiency, as these symmetric pairings do not at all scale well [GPS06]. There is no equivalent of the BDH problem defined on asymmetric pairings.

The construction of a symmetric pairing comes with a number of complexity implications.

Theorem 2.3. *The Discrete Log Problem in \mathbb{G}_1 or \mathbb{G}_2 is no harder than the Discrete Log Problem in \mathbb{G}_T .*

Proof. Consider $h_2 = g_2^a \in \mathbb{G}_2$, where $a = \log_P(Q)$ is unknown. We choose a random $g_1 \in \mathbb{G}_1$ and note that

$$e(g_1, h_2) = e(g_1, g_2^a) = e(g_1, g_2)^a.$$

Thus, also noting that e is easily computable, we have reduced the Discrete Log Problem in \mathbb{G}_2 to the Discrete Log Problem in \mathbb{G}_T . The proof is analogous for \mathbb{G}_1 , where we take a random $g_2 \in \mathbb{G}_2$. \square

Theorem 2.4. *The Decisional Diffie-Hellman Problem is easy in \mathbb{G}_1 .*

Proof. Given a tuple $\langle g, A = g^a, B = g^b, C = G^c \rangle$, a distinguisher can decide whether $c = ab$ by determining $v_1 = e(A, B) = e(g, g)^{ab}$ and $v_2 = e(g, C) = e(g, g)^c$. If $v_1 = v_2$ then $c = ab$ and the tuple is a DDH tuple. Otherwise, it is a non-DDH tuple. \square

Since the Decision Diffie-Hellman problem in \mathbb{G}_1 is easy, we cannot use DDH to build cryptosystems in these groups. Instead, there is a powerful variant of the Computational Diffie-Hellman assumption called the Bilinear Diffie-Hellman Assumption (BDH)[Jou00, BF01].

Definition 2.5 (BDH Problem). Let \mathbb{G}_1 and \mathbb{G}_T be groups of prime order q . Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_T$ be a pairing and let g be a generator of \mathbb{G}_1 . The *Bilinear Diffie-Hellman Problem* problem is as follows: Given $\langle g, g^a, g^b, g^c \rangle$ for some $a, b, c \in \mathbb{Z}_q^*$, compute $W = e(g, g)^{abc} \in \mathbb{G}_T$.

Parameters for the BDH problem can be generated by a *BDH parameter generator*. This is a randomized algorithm that takes security parameter κ and runs in polynomial time. It outputs a prime number q , the description of two groups \mathbb{G}_1 and \mathbb{G}_T of order q , and the description of a pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_T$.

The security parameter κ is used to determine the size of q ; for example, one could take q to be a random κ -bit prime. We assume that the description of the groups contain polynomial time algorithms for computing the group action and contains a generator. The generator enables us to generate uniformly random elements in \mathbb{G}_1 or \mathbb{G}_T . Similarly, we assume that the description of e contains a polynomial time algorithm for computing e .

An example of BDH parameter generation for the modified Weil pairing can be found in [BF01].

Definition 2.6 (BDH Assumption). The Bilinear Diffie-Hellman problem (BDH) is hard.

2.4 Relaxing Hashing Requirements

The Boneh-Franklin IBE scheme we will give in Section 4.2 uses a hash function $H : \{0, 1\}^* \mapsto \mathbb{G}_1^*$. A concrete IBE system uses \mathbb{G}_1 as a subgroup of the group of points on an elliptic curve. In practice, it is difficult to build hash functions that hash directly onto such groups. Therefore, rather than hash onto \mathbb{G}_1^* , we hash onto some set $A \subseteq \{0, 1\}^*$ and then use a deterministic encoding function $L : A \mapsto \mathbb{G}_1^*$ to map onto \mathbb{G}_1^* [BF01].

This encoding function must satisfy the following properties:

1. There must be an efficient deterministic algorithm to compute $L(x)$ for any $x \in A$.
2. For any $y \in \mathbb{G}_1^*$ the preimage of y under L has size exactly l . In other words, $|L^{-1}(y)| = l$ for all $y \in \mathbb{G}_1^*$. Note that this implies that $|A| = l \cdot |\mathbb{G}_1^*|$.
3. There is an efficient randomized algorithm \mathcal{L}_S such that \mathcal{L}_S induces a uniform distribution on $L^{-1}(y)$ for any $y \in \mathbb{G}_1^*$. In other words, $\mathcal{L}_S(y)$ is a uniform random element in $L^{-1}(y)$.

A specific encoding function for use with the modified Weil pairing can be found in [BF01].

A different way to implement H is the following construction from Waters [Wat05]. We choose a random value $u' \in \mathbb{G}_1$ and a random n -length vector $U = (u_i)$, whose elements are chosen at random from \mathbb{G}_1 .

Let v be an n -bit string representation of an identity, let v_i denote the i -th bit of v . Define the following function $H : \{0, 1\}^n \mapsto \mathbb{G}_1^*$:

$$H(v) = u' \prod_{i=1}^n u_i^{v_i}.$$

This function cannot directly be used as an instantiation of a random oracle (see Section 3.3) as it invalidates the proof. However, when replacing the random oracle with this function it is usually possible to prove security in the standard model, which makes it a very powerful tool.

Chapter 3

Security Notions

In this chapter we present our definition of security. We give a review of the different security models we will use and formalize each of them. We conclude this chapter with an explanation and a discussion of the random oracle model.

A convenient way to organize definitions of secure encryption is by considering separately the various possible *goals* and the various possible *attack models* and then formulate each definition as a pairing of a particular goal and a particular attack model. This model was suggested by Naor [Nao98]

By *goal* we mean the inability of an adversary to do something that we don't want him to do. For example, recovering the plaintext from a ciphertext. This is formalized by a game between a challenger and an adversary, where the adversary must try to achieve exactly what he should not be able to do. A proof of security can then be given by showing that a bounded adversary can only win with a non-negligible probability if he breaks some complexity assumption.

By *attack model* we mean the circumstances under which this game is played. This includes the information that the adversary has and the actions he may take. We formalize this by giving the adversary access to a collection of oracles. He may query these oracles to get additional information that can be used to win the game. An example is an oracle that can decrypt ciphertexts without knowing the secret key.

In the rest of this report, we say assume every adversary is polynomially bounded. We say that an adversary has an *advantage* in a game if he has a probability of winning the game that is non-negligibly higher than by just doing a random guess. An encryption is *secure* in one of these models if no adversary can gain an advantage against the challenger in the respective game.

We will now describe the several security models that we will use. We assume an asymmetric setting, where appropriate. At the end of Section 3.2 we will briefly discuss the symmetric setting.

3.1 Security Goals

First we discuss security goals of an encryption scheme. This part of the security model describes what an adversary should (not) be able to achieve. We consider the following goals:

Onewayness (OW) is an adversary's inability to completely recover the plaintext underlying a ciphertext.

Definition 3.1 (OW). An adversary chooses a public key pk . The challenger chooses a random message m and encrypts $c = \mathcal{E}_{pk}(m)$. The adversary tries to recover m .

A stronger notion is *Indistinguishability of encryptions* (IND), due to Goldwasser and Micali [GM84], which formalizes an adversary’s inability to learn any information about the plaintext underlying a ciphertext. They are formalized by the following games:

Definition 3.2 (IND). An adversary chooses a public key pk and two messages m_0 and m_1 , where $m_0 \neq m_1$. The challenger flips a coin $b \in \{0, 1\}$ and encrypts $c = \mathcal{E}_{pk}(m_b)$. The adversary tries to guess which message was encrypted.

Theorem 3.3 (IND \Rightarrow OW). *If there exists an encryption scheme Π that is secure in the sense of IND, then it is also secure in the sense of OW.*

Key privacy or anonymity (ANO), due to Bellare et al. [BBDP01], formalizes an adversary’s inability to learn any information about the public key pk used to encrypt a challenge ciphertext y .

Definition 3.4 (ANO). Let pk_0 and pk_1 be public keys, where $pk_0 \neq pk_1$. An adversary chooses a message m . The challenger flips a coin $b \in \{0, 1\}$ and encrypts $c = E_{pk_b}(m)$. The adversary tries to distinguish which key was used for the encryption.

Non-malleability (NM), due to Dolev et al. [DDN00], formalizes an adversary’s inability, given a ciphertext, to output a different ciphertext such that the plaintexts underlying these two ciphertexts are “meaningfully related”. We will not further discuss this goal, since we will not need it directly.

We also introduce a ‘new’ security goal, which is a combination of the two goals above. *Complete Indistinguishability* (CI) formalizes the adversary’s inability to learn any information about either the encrypted plaintext or the public key used to encrypt it.

Definition 3.5 (CI). An adversary chooses two messages m_0 and m_1 , and two public keys pk_0 and pk_1 , where $(pk_0, m_0) \neq (pk_1, m_1)$. The challenger flips a coin $b \in \{0, 1\}$ and encrypts $c = E_{pk_b}(m_b)$. The adversary makes a guess for b .

We will now prove that this game is equivalent with a combination of the IND and ANO games.

Lemma 3.6 (CI \Rightarrow IND \wedge ANO). *If there exists an encryption scheme Π that is secure in the sense of CI, then it is also secure in the sense of IND and in the sense of ANO.*

Proof. We first prove that CI \Rightarrow IND. Suppose there exists an adversary \mathcal{A} that can gain an advantage ϵ in winning an IND game. Then there exists an adversary \mathcal{B} that can gain an advantage in winning a CI game. \mathcal{B} works by simulating a challenger in an IND game with \mathcal{A} , with the same parameters as in the CI game with the challenger.

After \mathcal{A} chooses the messages m_0, m_1 and public key pk , \mathcal{B} gives the challenger m_0, m_1 and public keys $pk_0 = pk_1 = pk$. The challenger will flip a coin $b \in \{0, 1\}$ and encrypts challenge $c = \mathcal{E}_{pk_b}(m_b) = \mathcal{E}_{pk}(m_b)$. \mathcal{B} passes the challenge to \mathcal{A} , and repeats \mathcal{A} ’s guess b' as a guess to the challenger. Since \mathcal{A} has an advantage ϵ in guessing b correct, \mathcal{B} also has an advantage in guessing b correct.

However, since Π is secure in the sense of CI, such a \mathcal{B} does not exist. Thus \mathcal{A} does not exist and Π is secure in the sense of IND.

The proof that CI \Rightarrow ANO is analogous. □

Lemma 3.7 ($\text{IND} \wedge \text{ANO} \Rightarrow \text{CI}$). *If an encryption scheme Π is secure in the sense of both IND and ANO, then it is secure in the sense of CI.*

To complete this proof we will need a brid argument. In the proof we will create an algorithm \mathcal{B} , which will play two games with two different challengers at the same time. One game will be an IND game and the other will be a ANO game, both using the same scheme Π with the same parameters. This is depicted in Figure 3.1.

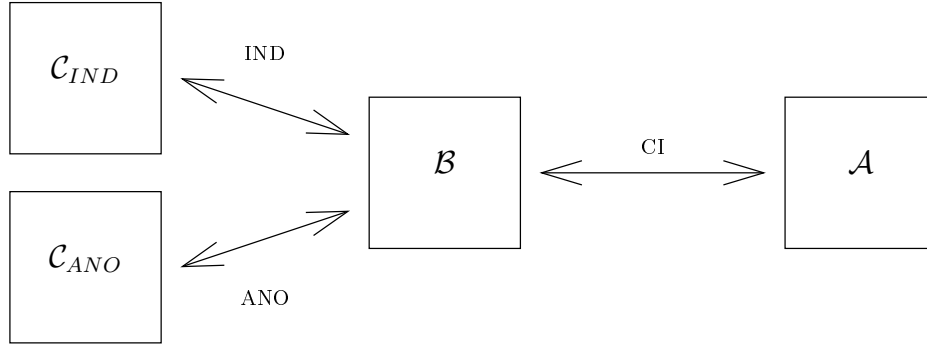


Figure 3.1: Simulator \mathcal{B} is an adversary in two games, while making use of \mathcal{A} .

We will then show that if there exists an adversary \mathcal{A} that has an advantage in a CI game targeted at Π , that \mathcal{B} has an advantage in winning either of these games. Since we assumed that Π is both secure in the sense of IND and in the sense of ANO, it should not be possible that \mathcal{B} has an advantage in winning either of these games. Thus we have a contradiction. We can state that \mathcal{A} cannot exist and thus that Π is CI secure.

We will use this same approach to proof Theorem 5.3 in Section 5.2.

Proof of Lemma 3.7. Suppose that Π is secure in the sense of IND and ANO. Assume there is an adversary \mathcal{A} that has nonnegligible advantage $\epsilon > 0$ in a CI game for Π . We show that there is an adversary \mathcal{B} that can achieve at least advantage $\epsilon/4$ in winning either an IND or an ANO game for Π .

Adversary \mathcal{B} works by simulating a challenger in a CI game with adversary \mathcal{A} . \mathcal{B} is supposed to give two messages and a public key to \mathcal{C}_{IND} and two public keys and a message to \mathcal{C}_{ANO} . She then receives a encryptions $c_{IND} = E_{pk}(m_a)$ from \mathcal{C}_{IND} and $c_{ANO} = E_{pk_b}(m)$ from \mathcal{C}_{ANO} , where a and b are random bits. \mathcal{B} outputs guesses a' and b' and wins if either $a' = a$ or $b' = b$. Note that when \mathcal{B} would do a random guess for both a' and b' , she has a probability of $3/4$ of winning the game.

\mathcal{B} interacts with \mathcal{A} as follows: \mathcal{A} outputs two pairs of a public key with a message; (m_0, pk_0) and (m_1, pk_1) . For now, we assume that $m_0 \neq m_1$ and $pk_0 \neq pk_1$. \mathcal{B} sends \tilde{m}_0, \tilde{m}_1 and pk_0 to \mathcal{C}_{IND} .

\mathcal{C}_{IND} returns a challenge $c_{IND} = E_{pk_0}(m_a)$; \mathcal{B} passes this challenge to \mathcal{A} . Note that this is only a valid challenge for the CI game when $a = 0$. Thus \mathcal{A} has advantage ϵ when $a = 0$ and no advantage when $a = 1$.

Finally \mathcal{A} gives a guess a' for a , \mathcal{B} repeats guess a' for the IND game and makes a random guess b' for b .

We can now compute the probability that \mathcal{B} wins either of the games:

$$\begin{aligned} \Pr[\mathcal{B} \text{ wins IND}] &= \Pr[a' = a] \\ &= \frac{1}{2} \Pr[a' = a \mid a = 0] + \frac{1}{2} \Pr[a' = a \mid a = 1] \\ &= \frac{1}{2} \left(\frac{1}{2} + \epsilon \right) + \frac{1}{2} \cdot \frac{1}{2} \\ &= \frac{1}{2} + \epsilon/2 \end{aligned}$$

$$\Pr[\mathcal{B} \text{ wins ANO}] = \Pr[b' = b] = \frac{1}{2}$$

$$\begin{aligned} \Pr[\text{Win either}] &= \Pr[a' = a \vee b' = b] \\ &= \Pr[a' = a] + \Pr[b' = b] - \Pr[a' = a] \Pr[b' = b] \\ &= \frac{1}{2} + \frac{1}{2} + \epsilon/2 - \frac{1}{2} \left(\frac{1}{2} + \epsilon/2 \right) \\ &= \frac{3}{4} + \epsilon/4 \end{aligned}$$

thus achieving an advantage of $\epsilon/4$.

We left out the case where $m_0 = m_1$; in this case \mathcal{B} can get an advantage of ϵ against \mathcal{C}_{ANO} and no advantage against \mathcal{C}_{IND} , to reach a total advantage of $\epsilon/2$ in winning either IND or ANO.

Likewise, when $pk_0 = pk_1$, she can get an advantage of ϵ against \mathcal{C}_{IND} and no advantage against \mathcal{C}_{ANO} to reach a total advantage of $\epsilon/2$ in winning either IND or ANO.

Thus \mathcal{B} has an advantage of at least $\epsilon/4$ in winning either a IND or ANO game against Π . This is in contradiction with the assumption that Π is both IND- and ANO-secure. This means that there can be no adversary \mathcal{A} , and thus Π is secure in the sense of CI. \square

Theorem 3.8 (CI \Leftrightarrow IND \wedge ANO). *If an encryption scheme Π is secure in the sense of both IND and ANO, then it is secure in the sense of CI. Vice versa, if Π is secure in the sense of CI, then it is secure in the sense of IND and secure in the sense of ANO.*

Proof. This follows directly from Lemmas 3.6 and 3.7. \square

3.2 Attack Models

Next, we consider different attacks. The difference between these is in the options that an adversary has during an attack. More formally, these attacks differ in the sort and amount of oracles an adversary has access to. An *oracle* is an abstract (and sometimes theoretical) source of information that the adversary can query to obtain information not otherwise available to him. Basically, an adversary with access to more powerful oracles can launch a more powerful attack.

A very weak setting is the *Chosen Plaintext Attack* (CPA). In this setting the attacker has access to an *encryption oracle*, which allows him to get the encryption of any plaintext

he wants. However, in a public key setting the adversary can already encrypt any plaintext himself if he knows the public key, making this attack the weakest attack possible.

A much stronger attack is the *Chosen Ciphertext Attack* (CCA) [NY90, RS91]. We now additionally give the adversary access to a *decryption* oracle. The decryption oracle decrypts any ciphertext for the adversary as long, so the adversary does not need the secret key. However, he may not ask the decryption of the challenge ciphertext. The adversary may query this oracle at any time, even after receiving the challenge.

Sometimes a difference is made between nonadaptive (CCA1) and adaptive (CCA2) models. In the weaker nonadaptive case, an adversary may only query the decryption oracle before he receives the challenge ciphertext. We only deal with the stronger CCA2 model, which we will just denote as CCA.

Example 3.9 (IND-CCA). Given public key pk , adversary \mathcal{A} chooses two message m_0 and m_1 . The challenger flips a coin $b \in \{0, 1\}$ and encrypts $c = E_{pk}(m_b)$. \mathcal{A} outputs a guess b' for b .

At any time \mathcal{A} may create different ciphertexts using pk . He may also at any time query the decryption oracle with a ciphertext $c' \neq c$ to find $m' = D_{sk}(c')$ (that is, before and after choosing m_0 and m_1 .)

As a next step, we want to adapt the model to the context of identity-based encryption¹ (ID). In this model we use a publicly available string (the identity) as public key, and generate an accompanying secret key. The big difference with a normal public key setting is the amount of keys. In a normal setting it is very normal that all public parameters are only used for one public/secret key pair. In identity-based encryption it is very normal to use the same public parameters for many identities at once.

So a new problem arises: it is possible that an adversary does not know the secret key for a specific encryption, but that he does know the secret key for one or more different identities. We do not want this to affect the security of the system in any way. We model this possibility by giving an adversary the choice of the identity he wishes to be challenged upon and giving him access to a *private key extraction* oracle. This is an oracle that an adversary can query with an identity; the oracle then returns a secret key that can decrypts ciphertext encrypted for this identity. The adversary can ask these queries at any time and in an adaptive way, but is obviously not allowed to ask a secret key that directly decrypts the challenge ciphertext. This captures the essence that an adversary knows any secret key except the one used for the challenge.

The literature sometimes makes a difference between *selective-id* (SID, [CHK03]) and *adaptive-id* (ID, [BF01]) identity-based models. In the weak selective-id case the adversary must choose the identity he wants to be challenged upon at the beginning of the game, before asking any other queries to oracles. In the stronger adaptive case the adversary declares the identity to be challenged upon at the time of the challenge and may ask any oracle queries before. We only deal with the adaptive setting.

Example 3.10 (ANO-ID-CPA). Adversary \mathcal{A} chooses a message m and two identities id_0 and id_1 . The challenger flips a coin $b \in \{0, 1\}$ and encrypts $c = E_{id_b}(m)$. \mathcal{A} outputs a guess b' for b .

At any time \mathcal{A} may query a private key extraction oracle to obtain the private key for identities $id' \notin \{id_0, id_1\}$ (that is, before and after choosing m , id_0 and id_1 .)

¹The concept of identity-based encryption is described in Chapter 4.

Finally, we use a completely different model for symmetric encryptions (SYM). In this setting we let the challenger choose a random symmetric key to encrypt the challenge. We do not give the adversary access to an encryption oracle.

Example 3.11 (IND-SYM). Adversary \mathcal{A} chooses two messages m_0 and m_1 . The challenger chooses a random symmetric key k , flips a coin $b \in \{0, 1\}$ and encrypts $c = E_k^{sym}(m_b)$. \mathcal{A} outputs a guess b' for b .

3.3 Random Oracle Model

A *random oracle* is a theoretical black box that acts like a random map. I.e. it responds to every query with a truly random response chosen uniformly from its output domain, except that for any specific query it responds the same way every time it receives that query.

Random oracles are a mathematical abstraction used in cryptographic proofs; they are typically used when no known implementable function provides the mathematical properties required by the proof. A system that is proven secure using such a proof is described as being secure in the *random oracle model*, as opposed to secure in the *standard model*. In practice, random oracles are typically used to model cryptographic hash functions in schemes where strong randomness assumptions are needed of the hash function's output.

In the more precise definition formalized by Bellare and Rogaway [BR93], the random oracle produces a bit-string of infinite length that can be truncated to the desired length. When a random oracle is used within a security proof, it is made available to all players, including the adversary. A single oracle may be treated as multiple oracle by prepending a fixed bitstring to the beginning of each query (e.g. queries formatted as “0|x” or “1|x” can be considered as calls to two separate random oracles.) Because we can transform an oracle in this way, we assume it is possible to have multiple random oracles at the same time with predetermined input and output domains.

It is not possible to implement a random oracle by a real function. In fact, certain protocols have been constructed that are proven secure in the random oracle model, but that are trivially insecure when any real hash function is substituted for the random oracle [CGH04]. Yet in spite of its shortcomings, the random oracle model seems to generate simple and efficient protocols against which no attacks are known. Consequently, the model can be regarded as a good initial idealized setting for designing and analyzing protocols. Still, it must be kept in mind that it does not, by itself, provide any security guarantees for implementations in the standard model. As such, it is only a *first step* towards a meaningful security analysis.

It is also interesting to note that one should be careful with the actual choice for implementing the random oracle. Standard hash functions are usually too structured to make a good random oracle. For example, consider MD5². [Tsu92] has observed that for any x there is a y such that for any z , MD5(xyz) can be easily computed given only $|x|$, MD5(x), and z .

This sort of structure does appear in applications and may possibly be exploited by an adversary, yet such predictable behavior does not break the requirements for a cryptographic hash function. Almost all standard hash functions suffer from being too structured to make good random oracles, but there are several constructs to make them. Among them are the following, or combinations of them:

²This argument holds apart from the fact that it has been broken now

1. A hash function with its output truncated or folded in some matter; e.g. $h_1(x) =$ the first 64 bits of $\text{SHA1}(x)$.
2. A hash function with its input lengths suitably restricted; e.g. $h_2(x) = \text{SHA1}(x)$, where $x \leq 400$.
3. A hash function used in some nonstandard way; e.g. $h_3(x) = \text{SHA1}(xx)$.

An example would be to settle on the (purely heuristic) choice of a map $h' : \{0, 1\}^{256} \mapsto \{0, 1\}^{64}$ defined by $h(x) =$ the first 64 bits of $\text{SHA}((xx) \oplus C)$, for a randomly chosen 512-bit constant C . This C could even be chosen at at instantiation time of the algorithm to ensure a “different” random oracle for every run.

A standard way to proof security in the random oracle is a proof by contradiction. First, the assumption that an adversary \mathcal{A} exists is made. Then it is shown that under this assumption it is also possible to construct an adversary \mathcal{B} that can solve some hard problem. Because the problem is hard \mathcal{B} can not exist, and thus \mathcal{A} can not exist.

Adversary \mathcal{B} works as a simulator by running internally a copy of the protocol and a copy of \mathcal{A} . It tries to make the internal protocol consistent with the hard problem and lets \mathcal{A} attack the simulated execution. It simulates in such a way that \mathcal{A} thinks that it observes a real-life execution. We can discern three different ways in which a simulator can handle the random oracle:

1. Handle the oracle as if controlled by a third party, where both the simulator \mathcal{B} and adversary \mathcal{A} get the identical responses to identical queries. In this case the simulator does not have any control over the output of the oracle, nor does it know which queries \mathcal{A} asks to the oracle.

This is intuitively the weakest model, in which we observe adversary \mathcal{A} as a black box.

2. Let simulator \mathcal{B} know the queries that \mathcal{A} asks to the random oracle. We call this model the *non-programmable random oracle* (NPRO) and it is inherent to viewing adversary \mathcal{A} as a white box, i.e. with the ability to view the internals of \mathcal{A} .

This model is usually used to model the *evaluation point knowledge* property of an adversary. One interpretation is that it is not possible to learn the value of $H(x)$ without knowing all of x .

3. Let simulator \mathcal{B} completely control the random oracle. Thus, \mathcal{B} has the possibility to adaptively change the answers to oracle queries as long as they are statistically indistinguishable from a truly random oracle.

We call this model the *programmable random oracle* (PRE), or just *random oracle* (RO). A justification is that the simulator chooses a mapping from the domain of all possible mappings for the random oracle, so from the adversary’s point of view there is no difference with a true random oracle.

While intuitively the first model is the weakest, it has the serious restriction that it only applies to black-box proofs. We think that this is too restrictive to model for practical protocols.

Nielsen [Nie02] showed that there exist protocols that can be shown to be secure in the third model, but which are impossible to prove secure in the second model, thus making a separation between these models.

In Section 4.2 the programmable random oracle is implicitly used for the proof of security of the Boneh-Franklin scheme. In Chapter 5 the non-programmable random oracle is used to proof security of the Fujisaki-Okamoto Transformation.

Chapter 4

Identity-Based Encryption

In this chapter we explain the concept of identity-based encryption. We follow with a formalization and a review of its applications and then treat the scheme introduced by Boneh and Franklin, along with some of its properties.

In 1984 Shamir [Sha84] asked for a public key encryption scheme in which the public key can be an arbitrary string. Shamir's original motivation for this encryption scheme was to simplify certificate management in e-mail systems. When Alice wants to send an e-mail to Bob at `bob@domain.com`, she simply encrypts her message using the public key string "`bob@domain.com`".

There is no need for Alice to obtain Bob's public key certificate. When Bob wants to decrypt the e-mail, he contacts a third party that we call the Private Key Generator (PKG). This PKG takes the same role as a certificate authority (CA) in regular public key encryption. Bob authenticates himself to the PKG and receives a private key. From then on he can use this private key to decrypt his messages.

Note that unlike in existing secure e-mail infrastructures, Alice can send encrypted email to Bob even when Bob does not yet have a secret key. Also note that key escrow is inherent in identity-based crypto-systems: the PKG knows Bob's private key.

In practice an identity-based scheme looks as follows: Let $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be an identity-based encryption scheme. Here

- $\mathcal{G}(\kappa)$ is the *Setup* algorithm, which takes security parameter κ and returns all public parameters required to use the scheme together with a *master key* s ,
- $\mathcal{K}(id)$ is the *Private Key extraction* algorithm, which takes an identity (string) id and uses master key s to produce a secret key sk matching id . Thus id/sk is a public/private key pair.
- $\mathcal{E}_{id}(m; r)$ is the *Encryption* algorithm, which takes a plaintext m and returns a ciphertext encrypted under identity id , where r is a random nonce,
- $\mathcal{D}_{sk}(c)$ is the *Decryption* algorithm, which takes a ciphertext c and returns the plaintext using secret key sk .

4.1 Applications

Aside from the original motivation for identity-based encryption, which is to help the deployment of a public key infrastructure, there are several other unrelated applications:

Revocation of public keys. PKI certificates contain a preset expiration date. In an IBE system key expiration can be done by having Alice encrypt e-mail to Bob using the public key "bob@company.com || 2007". In doing so Bob can use his private key during this year only. At the beginning of the next year Bob needs to obtain a new private key from the PKG.

Unlike existing PKI, Alice does not need to get a new certificate every time Bob gets a new private key. On the other hand, Bob cannot decrypt his e-mail if his private key is not renewed.

Managing user credentials. In a simple extension of the above, Alice can encrypt mail to Bob using the public key "bob@company.com || clearance=secret || 2007 ". Thus Bob can only read this mail if he has the required private key, which he only receives when he has secret clearance for the required date.

In this way it is easy to grant and revoke user credentials using the PKG. Alice does not have to check whether Bob has clearance and can be sure that he cannot read the mail if he does not. This is in contrast to traditional PKI, where this clearance would be in the certificate; Alice must check the certificate before encryption, but once she made the encryption, Bob will always be able to decrypt.

Delegation of duties. Suppose Alice encrypts mail to Bob using the subject as the public key. Bob can then decrypt the mail using the master key. Now, suppose that Bob has several assistants, each responsible for a different task (e.g. one is 'sales', another is 'administration', etc.).

Bob creates and gives a private key to each of his assistants, associated with the subject that is his responsibility. Each assistant can decrypt messages whose subject line falls within their responsibilities, but cannot decrypt messages intended for other assistants.

In this scenario Alice only obtains a single public key from Bob (the public parameters of the scheme) and she can use this public key to send mail with any subject line of her choice. The mail can then only be read by the assistant responsible, or Bob self.

Encryption with keyword search. Suppose Bob sends an encrypted email to Alice. This encrypted mail ends up on Alice's email gateway. However, Alice receives lots of mail, and she wants to receive only mail that has been flagged as 'urgent' directly on her phone.

Bob does this by adding an empty encryption to the mail, with public key 'urgent'. Alice can use the master key to create the secret key associated to 'urgent' and hand it to the mail gateway. When the gateway decrypts the addition and finds an empty encryption, it knows that the mail matches the search request.

4.2 Boneh-Franklin's `BasicIdent` scheme

The first identity-based encryption scheme that was both practical and secure was proposed by Boneh and Franklin [BF01]. Their solution makes novel use of groups for which there exists an efficiently computable bilinear map.

Their work consists of a basic scheme (`BasicIdent`) which they prove to be IND-ID-CPA secure. Then they use a specific form of the Fujisaki-Okamoto transformation (see Chapter 5) to transform this basic scheme into a IND-ID-CCA secure scheme. Later, Abdalla et. al. [ABC⁺05] prove that the `BasicIdent` scheme is also ANO-ID-CPA secure.

We will use the basic form (`BasicIdent`) of their scheme as a building block. This scheme is shown in Figure 4.1.

Setup $\mathcal{G}(\kappa)$: Using security parameter κ , generate BDH parameters to obtain $\mathbb{G}_1, \mathbb{G}_T$ of order q and an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_T$. Choose a generator $g \in \mathbb{G}_1$, pick a random $s \in \mathbb{Z}_q^*$ and let $g_{pub} = g^s$. Pick cryptographic hash functions $H_1 : \mathbb{G}_T \mapsto \{0, 1\}^n$ and $H_2 : \{0, 1\}^* \mapsto \mathbb{G}_1^*$ for some n , which will be viewed as random oracles in the security proof.

The messagespace is $\mathcal{M} = \{0, 1\}^n$, the ciphertext space is $\mathbb{G}_1 \times \{0, 1\}^n$. The master-key is s and the published public parameters are

$$\langle q, \mathbb{G}_1, \mathbb{G}_T, e, n, g, g_{pub}, H_1, H_2 \rangle.$$

Extract $\mathcal{K}(id)$: Given identity-string $id \in \{0, 1\}^*$, compute $Q = H_2(id)$ and $S = Q^s$. The private key is S .

Encryption $\mathcal{E}(m, id)$: To encrypt a message $m \in \mathcal{M}$, pick random $r \in \mathbb{Z}_q^*$ and compute $Q = H_2(id)$. The ciphertext is

$$C = \langle g^r, m \oplus H_1(e(Q, g_{pub})^r) \rangle.$$

Decryption $\mathcal{D}(C, S)$: Let $C = \langle U, V \rangle$ be a ciphertext created using the public key id . To decrypt C using private key S compute

$$m = V \oplus H_2(e(S, U)).$$

Figure 4.1: The Boneh-Franklin `BasicIdent` identity based encryption scheme

During encryption m is bitwise exclusive-ored with the hash of $e(Q, g_{pub})^r$. During decryption V is bitwise exclusive-ored with the hash of $e(S, U)$. The masks used during encryption and decryption are the same, since

$$H_2(e(S, U)) = H_2(e(Q^s, g^r)) = H_2(e(Q, g)^{rs}) = H_2(e(Q, g^s)^r) = H_2(e(Q, g_{pub})^r).$$

Thus, applying decryption after encryption produces the original message m as required.

This scheme is secure under chosen-plaintext attacks:

Theorem 4.1 ([BF01]). *Suppose that the hash functions H_1 and H_2 are random oracles. Then the `BasicIdent` scheme in Figure 4.1 is IND-ID-CPA secure assuming the BDH problem is hard relative to the generated BDH parameters.*

Concretely, suppose there is an IND-ID-CPA adversary \mathcal{A} that has advantage $\epsilon(\kappa)$ against the scheme *BasicIdent*. Suppose \mathcal{A} makes at most $q_E > 0$ private key extraction queries and $q_{H_2} > 0$ hash queries to H_2 . Then there is an algorithm \mathcal{B} that solves the BDH in the generated groups with advantage at least:

$$\text{Adv}(\kappa) \geq \frac{2\epsilon(\kappa)}{e(1 + q_E) \cdot q_{H_2}},$$

Here e is the base of the natural logarithm. The running time of \mathcal{B} is $O(\text{time}(\mathcal{A}))$.

Theorem 4.2 ([ABC⁺05]). Suppose that the hash functions H_1 and H_2 are random oracles. Then the *BasicIdent* scheme in Figure 4.1 is ANO-ID-CPA secure assuming that the BDH problem is hard relative to the generated BDH parameters.

We will later require a property of identity-based systems called γ -uniformity. This is a measure for how much the random coin chosen during encryption influences the ciphertext, e.g. it is $\gamma = 1$ for deterministic encryption and $\gamma = 2^{-q}$ for El Gamal encryption over \mathbb{Z}_q . In fact, if γ is not negligible then there can be no IND or ANO security.

Definition 4.3 (γ -uniformity). Let $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be an identity-based scheme with given master-key s . Define

$$\gamma_{id}(x, y) = \Pr[\exists r : y = \mathcal{E}_{id}(x; r)]$$

We say that Π is γ -uniform if $\gamma_{id}(x, y) \leq \gamma$ for any combination of x, y and id .

Lemma 4.4. The *BasicIdent* scheme is $\frac{1}{q-1}2^{-n}$ -uniform.

Proof. Let Π be a *BasicIdent* scheme with master-secret s . Then for any identity id , message x and ciphertext $y = (y_1, y_2)$

$$\begin{aligned} \gamma_{id}(x, y) &= \Pr[\exists r : y = \mathcal{E}_{id}(x; r)] \\ &= \Pr[\exists r : g^r = y_1 \wedge m \oplus H_1(e(H_2(id, g^s))^r) = y_2] \\ &= \Pr[\exists r : g^r = y_1] \Pr[m \oplus H_1(e(H_2(id, g^s))^r) = y_2 \mid g^r = y_1] \\ &= \frac{1}{q-1} \Pr[H_1(e(H_2(id, y_1))^s) = m \oplus y_2] \\ &\leq \frac{1}{q-1} 2^{-n} \end{aligned}$$

□

Chapter 5

Fujisaki-Okamoto Transformation

In [FO99], Fujisaki and Okamoto introduce a general method to convert an very weak OW-CPA secure asymmetric encryption scheme to a much stronger IND-CCA secure hybrid encryption scheme. Basically, this works by encrypting the plaintext with a symmetric cipher where the key is derived from an asymmetric encryption.

Let $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be an asymmetric encryption scheme. Formally, we denote by $E_{pk}(m; r)$ the encryption of m using random bits r under the public key pk . Fujisaki and Okamoto define the hybrid scheme

$$\mathcal{E}_{pk}^{fo}(m) = \left\langle \mathcal{E}_{pk}(\sigma; H(\sigma, m)) \| E_{G(\sigma)}^{sym}(m) \right\rangle,$$

where σ is generated at random, H and G are random oracles and E_k^{sym} indicates a IND-SYM secure symmetric cipher with key k . They show that the resulting scheme is IND-CCA secure in the random oracle.

In this chapter we will show that this transformation is also applicable to an identity-based encryption scheme. We give a new proof to show that we can use the same transformation to create CI-ID-CCA secure schemes, as defined in Chapter 3. In the rest of this chapter we will denote with $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ an identity-based scheme. With $\Pi^{fo} = (\mathcal{G}^{fo}, \mathcal{K}^{fo}, \mathcal{E}^{fo}, \mathcal{D}^{fo})$ we denote the Fujisaki-Okamoto transformation of Pi , which we define formally as follows:

\mathcal{G}^{fo} : in addition to the setup \mathcal{G} of Π , also set up two random oracles G and H , and choose a IND-SYM symmetric cipher E^{sym} .

\mathcal{K}^{fo} : identical to private key extraction algorithm \mathcal{K} .

\mathcal{E}^{fo} : Choose a random message σ . Then encrypt m as

$$\mathcal{E}_{id}^{fo}(m) = \left\langle \mathcal{E}_{id}(\sigma; H(\sigma, m)) \| E_{G(\sigma)}^{sym}(m) \right\rangle.$$

\mathcal{D}^{fo} : To decrypt a ciphertext $C = \langle U, V \rangle$ using secret key S , first decrypt $\sigma = \mathcal{D}_S(U)$. Then compute $m = D_{G(\sigma)}^{sym}(V)$. Finally check that the correct randomness was used: $\mathcal{E}_{id}(\sigma; H(\sigma, m)) = U$ ¹.

¹When we adapt this scheme to fuzzy encryption in Section 6.5, we will use an alternative method to check this.

5.1 A Weak Game

We begin by introducing a very weak new game, somewhere between a OW game and the CI game from Definition 3.5 in a chosen plaintext, identity-based setting. We will show that it defines a weaker model than CI-ID-CPA. Later in this chapter we will show that if an adversary cannot gain an advantage in this game under chosen plaintext attacks, we can use the Fujisaki-Okamoto transformation to construct a CI-ID-CCA secure scheme.

Definition 5.1 (Game-1). Adversary \mathcal{A} chooses two identities id_0 and id_1 . The challenger flips a coin $b \in \{0, 1\}$, chooses a random message $m \in \mathcal{M}$ and encrypts $c = \mathcal{E}_{id_b}(m)$. Later \mathcal{A} outputs a guess m' for m . \mathcal{A} wins if his guess is correct.

At any time \mathcal{A} may query a private key extraction oracle to obtain the private key for identities $id' \neq id_0, id_1$.

Note that when \mathcal{A} does a random guess he has a probability of $1/|\mathcal{M}|$ to correctly guess m . Thus \mathcal{A} has a probability of $1/|\mathcal{M}|$ to win the game.

Lemma 5.2. *Suppose there exists an adversary \mathcal{A} that can achieve an advantage ϵ in playing Game-1, while doing at most q_E public key extraction queries. Then there exists an adversary \mathcal{B} that can achieve advantage $\frac{1}{2}(\epsilon + 1/|\mathcal{M}|)$ in playing the CI-ID-CPA game, while doing at most q_E public key extraction queries.*

Proof. To proof this lemma we show how to construct adversary \mathcal{B} . Adversary \mathcal{B} works as a simulator that internally runs a copy of Game-1 and a copy of \mathcal{A} . A CI-ID-CPA game between \mathcal{B} and a challenger begins with the challenger doing any set-up and giving all public parameters to \mathcal{B} . \mathcal{B} may make private key extraction queries at any time and must eventually select two identities id_0, id_1 and two messages m_0, m_1 . The challenger flips a random coin $b \in \{0, 1\}$ and encrypts the challenge $c = \mathcal{E}_{id_b}(m_b)$. Finally, \mathcal{B} must make a guess b' for b .

\mathcal{B} internally starts a Game-1 game with \mathcal{A} . She passes all public information she got from the challenger to \mathcal{A} . She relays all public key extraction queries to the challenger. When \mathcal{A} is ready for the challenge, he chooses two identities id_0, id_1 and sends them to \mathcal{B} . \mathcal{B} then selects two random messages m_0, m_1 . She sends id_0, id_1, m_0 and m_1 to the challenger.

The challenger flips a random coin $b \in \{0, 1\}$ and responds with a challenge $\mathcal{E}_{id_b}(m_b)$. \mathcal{B} relays this challenge to \mathcal{A} . Eventually \mathcal{A} comes with a guess m' for m_b . If possible, \mathcal{B} does a guess b' , such that $m_{b'} = m'$. Otherwise, she outputs a random guess b' .

We will now analyze \mathcal{B} . Define the following events

SuccB: \mathcal{B} wins the CI-ID-CPA game, i.e. $b = b'$,

SuccA: \mathcal{A} wins Game-1, i.e. $m' = m_b$, $\Pr[\text{SuccA}] = \frac{1}{|\mathcal{M}|} + \epsilon$.

We can now compute

$$\begin{aligned} \Pr[\text{SuccB}] &= \Pr[\text{SuccB} \mid \text{SuccA}] \Pr[\text{SuccA}] + \Pr[\text{SuccB} \mid \neg\text{SuccA}] \Pr[\neg\text{SuccA}] \\ &\geq \left(\frac{1}{|\mathcal{M}|} + \epsilon \right) + \frac{1}{2} \left(1 - \frac{1}{|\mathcal{M}|} - \epsilon \right) \\ &= \frac{1}{2} + \frac{1}{2} \left(\epsilon + \frac{1}{|\mathcal{M}|} \right). \end{aligned}$$

Thus, \mathcal{B} has an advantage at least $\frac{1}{2}(\epsilon + 1/|\mathcal{M}|)$

□

Thus we can conclude that a CI-ID-CPA secure scheme is also secure in the sense of Game-1. Thus any of the following results that hold for a Game-1 secure scheme also hold for CI-ID-CPA secure schemes.

Corollary. *If a scheme is secure in the sense of CI-ID-CPA, then it is also secure in the sense of Game-1.*

5.2 Security Under Chosen Plaintext Attacks

We will now prove that application of the Fujisaki-Okamoto transformation to a scheme that is secure in the sense of Game-1 (Definition 5.1) will result in a scheme that is secure in the sense of CI-ID-CPA. Consequently, application of Fujisaki-Okamoto to a CI-ID-CPA secure scheme will result in a scheme that is also CI-ID-CPA. In the next section we will show that this scheme is also secure in a chosen ciphertext setting.

To complete this proof we will need a hybrid argument. In the proof we will create an algorithm \mathcal{B} , which will play two games with two different challengers at the same time. One game will be the weak Game-1 that we introduced in the previous section, targeted at the identity-based scheme Π . The other game will be an IND-SYM game, targeted at the symmetric cipher E^{sym} . This is depicted in Figure 5.1.

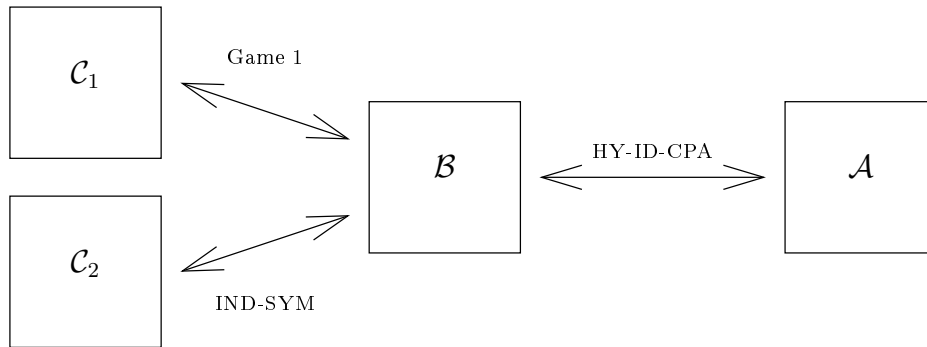


Figure 5.1: Simulator \mathcal{B} is an adversary in two games, while making use of \mathcal{A} .

We will then show that if there exists an adversary \mathcal{A} that has an advantage in a CI-ID-CPA game targeted at Π^{fo} , that \mathcal{B} has an advantage in winning either of these games. Since we assumed that Π is secure in the sense of Game-1 and that E^{sym} is secure in the sense of IND-SYM, it should not be possible that \mathcal{B} has an advantage in winning either of these games. Thus we have a contradiction. We can state that \mathcal{A} cannot exist and thus that Π^{fo} is CI-ID-CPA secure.

The proof relies mostly on the following concept: in the proof, the challengers choose some random parameters that the simulator does not know. This means that in practice the simulator \mathcal{B} would not be able to create a valid ciphertext for adversary \mathcal{A} , because she does not know how to answer a few specific oracle queries. If \mathcal{A} does not ask these queries but still manages to give an answer, he apparently did not need these queries. Because the oracle is a truly random map, it is from \mathcal{A} 's perspective still possible that \mathcal{B} would give the right answers to these queries. Thus the challenge is a valid one and \mathcal{A} has an advantage that \mathcal{B} can use in the IND-SYM game.

On the other hand, if \mathcal{A} does ask these specific queries the ciphertext becomes invalid. This is because \mathcal{B} will have to guess the random parameters chosen by the challengers and this guess is with a large probability incorrect. Because the ciphertext is incorrect, \mathcal{A} will have no advantage. However, \mathcal{A} can only ask these specific queries if he knows the message contained in the challenge of Game-1. This means that \mathcal{B} gets an advantage in playing Game-1.

This leads us to the following theorem. The bounds for the advantage are not strict, meaning that the transformation does not always work. However, as long as the amount of oracle queries does not exceed the size of the messagespace \mathcal{M} , there is a positive advantage for \mathcal{B} . Practically, that would be a brute-force attack.

Theorem 5.3. *Let \mathcal{A} be an adversary that has advantage ϵ against scheme Π^{fo} in an CI-ID-CPA game. Suppose \mathcal{A} makes at most $q_E \geq 0$ private key extraction queries, q_G queries to random oracle G and q_H queries to random oracle H , with $(q_G + q_H) < |\mathcal{M}|$. Then there is an adversary \mathcal{B} that can achieve an advantage in winning either a Game-1 game against Π or a IND-SYM game against E^{sym} , while doing at most the same amount of queries. It's running time is equal to the running time of \mathcal{A} .*

Proof. The games between \mathcal{C}_1 , \mathcal{C}_2 and \mathcal{B} start with the challengers doing all the required set-up and passing the acquired public parameters to \mathcal{B} . \mathcal{B} is supposed to give two identities id_0, id_1 to \mathcal{C}_1 and two messages m_0 and m_1 to \mathcal{C}_2 to be challenged upon. She then receives an asymmetric encryption of a random message σ under identity id_a ($a \in \{0, 1\}$) from \mathcal{C}_1 and a symmetric encryption of m_b ($b \in \{0, 1\}$) under a random key k from \mathcal{C}_2 . \mathcal{B} outputs a guess m' for m to \mathcal{C}_1 and a guess b' for b to \mathcal{C}_2 .

\mathcal{B} simulates the random oracles H and G by preparing empty lists G^{list} and H^{list} , and a counter $i = 1$. She then answers oracle queries as follows:

Queries for $G(\sigma)$: If there is a tuple (σ_j, g_j) in G^{list} with $\sigma_j = \sigma$, then return g_j . Otherwise choose random g , set $(\sigma_i, g_i) := (\sigma, g)$, increase i with 1, and return g .

Queries for $H(\sigma, m)$: If there is a tuple (σ_j, m_j, h_j) in H^{list} with $\sigma_j = \sigma$ and $m_j = m$, then return h_j . Otherwise choose random h , set $(\sigma_i, m_i, h_i) := (\sigma, m, h)$, increase i with 1, and return h .

Private key extraction: $\mathcal{K}^{fo}(id)$: \mathcal{B} relays these queries to challenger \mathcal{C}_1 , i.e. she answers $\mathcal{K}^{fo}(id) = \mathcal{K}(id)$.

\mathcal{B} gives \mathcal{A} all the public parameters and access to the oracle H and G .

Eventually \mathcal{A} will output two identities id_0, id_1 and two messages m_0, m_1 . \mathcal{B} will give id_0 and id_1 to \mathcal{C}_1 . She will give m_0 and m_1 to \mathcal{C}_2 . \mathcal{C}_1 will choose a random message σ , an random coin $a \in \{0, 1\}$, a random nonce r , and then encrypt challenge $c_1 = \mathcal{E}_{id_a}(\sigma; r)$. \mathcal{C}_2 will choose a random $b \in \{0, 1\}$, a random key k and then encrypt challenge $c_2 = E_k^{sym}(m_b)$. Note that \mathcal{B} does not know m , r , a , b and k .

\mathcal{B} constructs $C = \langle c_1 || c_2 \rangle$ and gives it as a challenge to \mathcal{A} . Note that this is a valid ciphertext of message m_0 under identity id_0 or of message m_1 under identity id_1 only when $a = b$, $r = H(\sigma, m_b)$ and $k = G(\sigma)$. \mathcal{A} will output a guess b' for b . \mathcal{B} relays guess b' to \mathcal{C}_2 . She will then pick a random either a tuple (σ_j, g_j) from G^{list} or a tuple (σ_j, m_j, h_j) from H^{list} and output $\sigma' = \sigma_j$ as a guess to \mathcal{C}_1 .

We will now proceed to analyze \mathcal{B} . We here define the following events:

Equal a chosen by \mathcal{C}_1 is equal to b chosen by \mathcal{C}_2 ,

Ask \mathcal{A} queries G for $G(\sigma)$ or H for $H(\sigma, m_b)$,

SuccA \mathcal{A} wins the IND-ID-CPA game,

SuccB1 \mathcal{B} wins the Game-1 game with \mathcal{C}_1 , i.e. $\sigma' = \sigma$,

SuccB2 \mathcal{B} wins the IND-SYM game with \mathcal{C}_2 , i.e. $b' = b$,

SuccB \mathcal{B} wins either the Game-1 game with \mathcal{C}_1 or the IND-SYM game with \mathcal{C}_2 .

First suppose \neg **Equal**, which happens with probability $1/2$. Thus $a \neq b$ and the challenge C is definitely not a valid challenge for the CI-ID-CPA game. This means that \mathcal{A} 's best guess will be a random guess. In that case \mathcal{B} has a probability of $1/2$ of winning the IND-SYM game and a probability of $1/|\mathcal{M}|$ of winning Game-1. Thus \mathcal{B} has no advantage:

$$\Pr[\text{SuccB} \mid \neg\text{Equal}] = \frac{1}{2} + \frac{1}{|\mathcal{M}|}$$

Now suppose that **equal** occurs and that at some point \mathcal{A} asks a query $G(\sigma)$ to G or $H(\sigma, m_b)$ to H (event **Ask**). Then C is probably not a valid cipher text for the CI-ID-CPA game, because \mathcal{B} does not know r and k and can not make sure that $H(\sigma, m_b) = r$ and $G(\sigma) = k$. Thus \mathcal{A} 's best guess is a random guess and

$$\Pr[\text{SuccB2} \mid \text{Ask} \wedge \text{Equal}] = \frac{1}{2}.$$

However, then σ will occur in either G^{list} or H^{list} , which gives \mathcal{B} an advantage in the Game-1 game:

$$\Pr[\text{SuccB1} \mid \text{Ask} \wedge \text{Equal}] \geq (q_G + q_H)^{-1}$$

Finally suppose that **equal** occurs and that \mathcal{A} does not ask such a query (event \neg **Ask**). Then from \mathcal{A} 's point of view C is a correct ciphertext of either m_0 under identity id_0 or m_1 under identity id_1 , thus keeping his advantage. Since \mathcal{A} can tell whether m_0 or m_1 was encrypted in C , \mathcal{B} can tell which one is in c_2 . Thus,

$$\Pr[\text{SuccB2} \mid \neg\text{Ask} \wedge \text{Equal}] = \Pr[\text{SuccA} \mid \neg\text{Ask} \wedge \text{Equal}] = \frac{1}{2} + \epsilon$$

However, there is no guarantee that σ is in either G^{list} or H^{list} , thus

$$\Pr[\text{SuccB1} \mid \neg\text{Ask} \wedge \text{Equal}] = 1/|\mathcal{M}|.$$

Taking these all together and given that the games with \mathcal{C}_1 and \mathcal{C}_2 are completely independent we get

$$\begin{aligned} \Pr[\text{SuccB1} \mid \text{Equal}] &= \Pr[\text{SuccB1} \mid \text{Equal} \wedge \text{Ask}] \Pr[\text{Ask}] + \Pr[\text{SuccB1} \mid \text{Equal} \wedge \neg\text{Ask}] (1 - \Pr[\text{Ask}]) \\ &\geq \frac{\Pr[\text{Ask}]}{q_G + q_H} + \frac{1 - \Pr[\text{Ask}]}{|\mathcal{M}|} \\ &= \left(\frac{1}{q_H + q_H} - \frac{1}{|\mathcal{M}|} \right) \Pr[\text{Ask}] + \frac{1}{|\mathcal{M}|} \end{aligned}$$

$$\begin{aligned}
\Pr[\text{SuccB2} \mid \text{Equal}] &= \Pr[\text{SuccB2} \mid \text{Equal} \wedge \text{Ask}] \Pr[\text{Ask}] + \Pr[\text{SuccB2} \mid \text{Equal} \wedge \neg\text{Ask}] (1 - \Pr[\text{Ask}]) \\
&= \frac{1}{2} \Pr[\text{Ask}] + \left(\frac{1}{2} + \epsilon\right) (1 - \Pr[\text{Ask}]) \\
&= \frac{1}{2} + \epsilon(1 - \Pr[\text{Ask}])
\end{aligned}$$

$$\begin{aligned}
\Pr[\text{SuccB} \mid \text{Equal}] &= \Pr[\text{SuccB1} \mid \text{Equal}] + \Pr[\text{SuccB2} \mid \text{Equal}] \\
&\geq \left(\frac{1}{q_H + q_H} - \frac{1}{|\mathcal{M}|}\right) \Pr[\text{Ask}] + \frac{1}{|\mathcal{M}|} + \frac{1}{2} + \epsilon(1 - \Pr[\text{Ask}]) \\
&= \frac{1}{2} + \frac{1}{|\mathcal{M}|} + \epsilon + \Pr[\text{Ask}] \left(\frac{1}{q_G + q_H} - \frac{1}{|\mathcal{M}|} - \epsilon\right)
\end{aligned}$$

Which gives the following probability of success:

$$\begin{aligned}
\Pr[\text{SuccB}] &= \frac{1}{2} \Pr[\text{SuccB} \mid \text{Equal}] + \frac{1}{2} \Pr[\text{SuccB} \mid \neg\text{Equal}] \\
&= \frac{1}{2} \cdot \left(\frac{1}{2} + \frac{1}{|\mathcal{M}|}\right) + \frac{1}{2} \cdot \left(\frac{1}{2} + \frac{1}{|\mathcal{M}|} + \epsilon + \Pr[\text{Ask}] \left(\frac{1}{q_G + q_H} - \frac{1}{|\mathcal{M}|} - \epsilon\right)\right) \\
&= \frac{1}{2} + \frac{1}{|\mathcal{M}|} + \left[\frac{\epsilon}{2} + \frac{1}{2} \Pr[\text{Ask}] \left(\frac{1}{q_G + q_H} - \frac{1}{|\mathcal{M}|} - \epsilon\right)\right]
\end{aligned}$$

If $\frac{1}{q_G + q_H} - \frac{1}{|\mathcal{M}|} - \epsilon \geq 0$ then

$$\Pr[\text{SuccB}] \geq \frac{1}{2} + \frac{1}{|\mathcal{M}|} + \frac{\epsilon}{2},$$

giving \mathcal{B} an advantage of $\frac{\epsilon}{2}$. On the other hand, if we have that $\frac{1}{q_G + q_H} - \frac{1}{|\mathcal{M}|} - \epsilon < 0$ we get an advantage of

$$\begin{aligned}
&\frac{\epsilon}{2} + \frac{1}{2} \left(\frac{1}{q_G + q_H} - \frac{1}{|\mathcal{M}|} - \epsilon\right) \\
&= \frac{1}{2} \left(\frac{1}{q_G + q_H} - \frac{1}{|\mathcal{M}|}\right),
\end{aligned}$$

which is strictly positive as long as $(q_G + q_H) < |\mathcal{M}|$, i.e. as long as the amount of oracle queries to G and H is not larger than the size of the message-space.

The running time of \mathcal{B} is equal to the running time of \mathcal{A} .

□

5.3 Chosen Ciphertext Attacks

We will now show that the cryptosystem we obtained by using the Fujisaki-Okamoto transformation is not only secure under chosen plaintext attacks, but also under chosen ciphertext attacks. Since the only difference between these games is the presence of a decryption oracle, we must find a way to add this oracle to the simulator from the last section.

The problem is that the simulator can not just decrypt any message, since she does not know the required secret keys. She does not have access to a decryption oracle herself, so she must find some other way to find the contents of a ciphertext. To solve this problem end we use a concept by Bellare and Rogaway [BR94, BDPR98], and also used by [FO99] in their proof. It is based on a *knowledge extractor*, an algorithm that finds the contents of a ciphertext by looking at all the queries that have been done to the various random oracles.

Lemma 5.4. *Let $\Pi = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be an γ -uniform IBE-scheme and let $\Pi^{fo} = (\mathcal{G}^{fo}, \mathcal{K}^{fo}, \mathcal{E}^{fo}, \mathcal{D}^{fo})$ be its Fujisaki-Okamoto transformation. Suppose there exists an CI-ID-CCA adversary \mathcal{A} against Π^{fo} that can achieve advantage ϵ while doing at most q_E private key extraction queries, q_D decryption queries, q_G queries to random oracle G and q_H queries to random oracle H . Then there is an CI-ID-CPA adversary \mathcal{B} against Π^{fo} that can achieve advantage $\epsilon(1 - q_D(\gamma - \frac{1}{|G|}) - \frac{q_D}{2}(\gamma + \frac{1}{|G|}))$, where $|G|$ is the size of the range of G .*

Proof. The game between the challenger and the adversary \mathcal{B} start with the challenger running the setup-algorithm \mathcal{G} and publishing all public parameters. \mathcal{B} is supposed to choose two identities id_0, id_1 and output two messages m_0, m_1 and expects an encryption $c = E_{id_b}(m_b)$ for random $b \in \{0, 1\}$. Eventually \mathcal{B} does a guess $b' \in \{0, 1\}$ for b . At any time she may query a private key extraction oracle and random oracles G^c and H^c at any time. Adversary \mathcal{B} works by interacting with \mathcal{A} in an CI-ID-CCA game as follows:

\mathcal{B} relays all public parameters to \mathcal{A} to set up the CI-ID-CCA game. She prepares two empty lists G^{list} and H^{list} , and a counter $i = 1$. She gives \mathcal{A} access to a private key extraction oracle, a decryption oracle, as well as random oracles G and H , to which she replies as follows

Private Key Extraction: She passes every query for a private key directly to the challenger.

Queries for $G(\sigma)$: If there is a tuple (σ_j, g_j) in G^{list} with $\sigma_j = \sigma$, then return g_j . Otherwise $g = G^c(\sigma)$, set $(\sigma_i, g_i) := (\sigma, g)$, increase i with 1, and return g .

Queries for $H(\sigma, m)$: If there is a tuple (σ_j, m_j, g_j) in H^{list} with $\sigma_j = \sigma$ and $m_j = m$, then return h_j . Otherwise $h = H^c(\sigma, m)$, set $(\sigma_i, m_i, h_i) := (\sigma, m, h)$, increase i with 1, and return h .

Decryption Queries: Upon receiving a decryption query for identity id and ciphertext $C = (c1, c2)$ do the following:

1. Set two empty lists, S_1 and S_2 .
2. Find all elements in H^{list} such that $c_1 = \mathcal{E}_{id}(\sigma_i; h_i)$ and put them in list S_1 . If $S_1 = \emptyset$, output \perp (for ‘invalid encryption’) and stop.
3. For every (σ_i, m_i, h_i) in S_1 , find all elements (σ_j, g_j) in G^{list} such that $\sigma_i = \sigma_j$ and put $(\sigma_i, m_i, h_i) \parallel (\sigma_j, g_j)$ into S_2 . If $S_2 = \emptyset$, output \perp and stop.
4. Check in S_2 if there is a tuple $(\sigma_i, m_i, h_i) \parallel (\sigma_j, g_j)$ such that $c_2 = E_{g_j}^{sym}(m_i)$. If it exists in S_2 , then output m_i . Otherwise, output \perp .

Eventually \mathcal{A} will output two identities id_0, id_1 and two messages x_0 and x_1 . \mathcal{B} will give these to the challenger. The challenger flips a coin $b \in \{0, 1\}$ and returns a challenge $C = \langle \mathcal{E}_{id_b}(\sigma; r), E_k^{sym}(x_b) \rangle$, for random σ . Here $r = H^c(\sigma, x_b, id)$ and $k = G^c(\sigma, id)$; \mathcal{B} gives this challenge to \mathcal{A} . When \mathcal{A} outputs a guess b' for b , then \mathcal{B} also outputs guess b' for b .

To analyze the decryption oracle we define the following events:

$p(S_1)$ is true if $S_1 \neq \emptyset$,

$p(S_2)$ is true is $S_2 \neq \emptyset$,

Find is true if there exists a $(\sigma_i, m_i, id_i, h_i) || (\sigma_j, id_j, g_j)$ in S_2 such that $c_2 = E_{g_j}^{sym}(m_i)$,

Fail is true if “the output of the decryption oracle for ciphertext $C = (c_1, c_2)$ and identity id is not equal to $\mathcal{D}_{id}^{fo}(C)$. Note that the oracle does not fail when it outputs \perp on an invalid ciphertext.

SuucA is true if \mathcal{A} wins the CI-ID-CCA game.

SuccB is true if \mathcal{B} wins the CI-ID-CPA game.

We also introduce the following events:

0 $\neg p(S_1)$

10 $p(S_1) \wedge \neg p(S_2)$

110 $p(S_1) \wedge p(S_2) \wedge \neg \text{Find}$

111 $p(S_1) \wedge p(S_2) \wedge \text{Find}$

The following equations holds:

$$\begin{aligned} \Pr[\text{Fail}] &= \Pr[0] \Pr[\text{Fail} \mid 0] + \Pr[10] \Pr[\text{Fail} \mid 10] + \Pr[110] \Pr[\text{Fail} \mid 110] + \Pr[111] \Pr[\text{Fail} \mid 111] \\ &\leq \Pr[\text{Fail} \mid 0] + \Pr[\text{Fail} \mid 10] + \Pr[\text{Fail} \mid 110] + \Pr[\text{Fail} \mid 111] \end{aligned}$$

Here we can easily see that by construction

$$\Pr[\text{Fail} \mid 110] = 0 \quad \text{and} \quad \Pr[\text{Fail} \mid 111] = 0.$$

Claim. $\Pr[\text{Fail} \mid 0] \leq \gamma (= \frac{1}{q})$

Proof. In the case of 0, the decryption oracle fails when (c_1, c_2) actually was a valid ciphertext for the given identity id . That is, if there exist a σ and a h such that $c_1 = \mathcal{E}_{id}(\sigma; h)$. Thus, because of the γ -

$$\begin{aligned} \Pr[\text{Fail} \mid 0] &= \Pr[\exists(\sigma, h) \mid c_1 = \mathcal{E}_{id}(\sigma; h)] \\ &= \sum_{\sigma} \Pr[\sigma] \Pr[\exists h \mid c_1 = \mathcal{E}_{id}(\sigma; h)] \\ &\leq \sum_{\sigma} \Pr[\sigma] \gamma \\ &= \gamma \end{aligned}$$

□

Claim. $\Pr[\text{Fail} \mid 10] \leq \frac{1}{|G|}$

Proof. Suppose that $C = (c_1, c_2)$ is a valid ciphertext for identity id . In the case 10 there exists a tuple $(\sigma_i, m_i, h_i) \in H^{list}$ such that $c_1 = \mathcal{E}(\sigma_i, h_i)$, but there is no $(\sigma_j, g_j) \in G^{list}$, i.e. the adversary never requested $G(\sigma)$. This happens when the adversary did a good guess for c_2 ; thus there exists a g such that $c_1 = E_g^{sym}(\sigma_i)$.

$$\Pr[\text{Fail} \mid 10] = \Pr[\exists g \mid E_g^{sym}(\sigma_i) = c_2] \leq \frac{1}{|G|}$$

□

Adversary \mathcal{B} has the same chance of winning as adversary \mathcal{A} , except that for every query to the decryption oracle there is a chance $\Pr[\text{Fail}]$ that the decryption oracle fails. From the claims above we can derive that

$$\begin{aligned} \Pr[\text{Fail}] &\leq \Pr[\text{Fail} \mid 0] + \Pr[\text{Fail} \mid 10] + \Pr[\text{Fail} \mid 110] + \Pr[\text{Fail} \mid 111] \\ &\leq \gamma + \frac{1}{|G|} \end{aligned}$$

$$\begin{aligned} \Pr[\text{SuccB} \mid \text{SuccA}] &= 1 - q_D \Pr[\text{Fail}] \\ &\geq 1 - q_D \left(\gamma + \frac{1}{|G|} \right) \end{aligned}$$

$$\begin{aligned} \Pr[\text{SuccB}] &= \Pr[\text{SuccA}] \Pr[\text{SuccB} \mid \text{SuccA}] \\ &\geq \left(\frac{1}{2} + \epsilon \right) \left(1 - q_D \left(\gamma + \frac{1}{|G|} \right) \right) \\ &= \frac{1}{2} + \left(\epsilon \left(1 - q_D \left(\gamma + \frac{1}{|G|} \right) \right) - \frac{q_D}{2} \left(\gamma + \frac{1}{|G|} \right) \right) \end{aligned}$$

□

Note. This is not a tight reduction and \mathcal{B} 's advantage is dependent on the parameters. However, $\frac{1}{|G|}$ and γ are usually so small that the factors $q_D \gamma$ and $q_D / |G|$ are negligible compared to ϵ , thus resulting in a positive advantage. From here we assume that this is indeed the case, which is generally accepted in literature.

Theorem 5.5. *Let Π^{fo} be the Fujisaki-Okamoto transform of a γ -uniform IBE scheme Π . Let \mathcal{A} be an adversary ϵ against scheme Π^{fo} in an CI-ID-CCA game. Suppose \mathcal{A} makes at most $q_E \geq 0$ private key extraction queries, $q_G \geq 0$ queries to random oracle G and $q_H \geq 0$ queries to random oracle H , with $(q_G + q_H) < |\mathcal{M}|$. Then there is an adversary \mathcal{B} that can achieve an advantage in winning either a Game-1 game against Π , or a IND-SYM game against E^{sym} , while doing at most the same amount of queries.*

Proof. This follows directly from Theorem 5.3 and Lemma 5.4. □

Chapter 6

Fuzzy Identity-Based Encryption

6.1 Introduction

In this chapter we will construct the actual anonymous fuzzy identity-based encryption scheme, using all the building blocks from the previous chapters.

We consider a user's biometric, for example an iris scan, as that user's identity described by several attributes and then encrypt to the user using their biometric identity, which we consider as public information. We could use a regular identity-based scheme, as described in Chapter 4, where we use a string containing all these attributes as the public key.

Biometric measurements are noisy however, and a single bit of difference between the measurement used for encryption and the measurement used to generate the secret key would result in the receiver being unable to decrypt his messages. This means that if we want to use an identity-based encryption system in this way, it requires some sort of tolerance for errors in the measurement.

So what kind of errors do we tolerate? Informally, a secret key should be able to decrypt an encryption if the biometric measurement used to create it belongs to the same person as the biometric measurement used to encrypt the message. If a biometric measurement is good, then measurements of one person will be closely related, while measurements of different persons will be completely different.

In practice we will view biometric information as a set of specific attributes and take the Hamming distance between these sets as a metric: if the Hamming distance between two biometric measurements is below a certain threshold we will assume it is from the same person; if the Hamming distance is above this threshold then we will assume they are from different persons.

Let v be some person's identity, e.g. $v = \text{'Bob Smith'}$. This person gets his fingerprint taken, which we call ω . We shall assume that this fingerprint is vector l attributes, where each attribute may be a bitstring, thus $\omega = (\omega_1, \dots, \omega_l)$.

At some other moment he can have his fingerprint taken again. We will call this measurement ω' . Since the process is noisy, $\omega \neq \omega'$. The measurements do belong to the same person however, so we assume that the Hamming distance $d(\omega, \omega') \leq t$, for a certain threshold t .

Now let's take another fingerprint, ω'' , which does not belong to Bob, but to a different identity v'' . We then assume that $d(\omega, \omega'') > t$ and $d(\omega', \omega'') > t$. This allows us to distinguish between the same person or different persons by looking at the Hamming distance.

In this chapter we will construct a basic fuzzy identity-based encryption scheme on this

technique. We will then show it is secure under chosen plaintext attacks in the sense of both anonymity and indistinguishability. Finally we will use Fujisaki-Okamoto to transform it to a scheme that is secure under chosen ciphertext attacks.

6.2 Basic Scheme

We will now introduce a very simple scheme based on this threshold. The basic idea is that an encrypted message can only be recovered if at least $l - t$ of the attributes in the biometric match. This leads to a Shamir $(l - t, l)$ -threshold scheme [Sha84].

Again, let id be an identity and $\omega = (\omega_1, \dots, \omega_l)$ be its biometric. Let m be the message we want to encrypt. We split this message into l shares using a $(l - t, l)$ -threshold scheme, so that at least $l - t$ of the share are needed to reconstruct the message. Or in other words: at most t can get lost to still reconstruct the message.

Next we will use the **BasicIdent** scheme from Figure 4.1 to encrypt these shares. To encrypt the i -th share we will use ω_i , the i -th part of v 's biometric, as a public key. Obviously, the receiver will need the secret key for at least $l - t$ of the ω_i . If the receiver has less, then he can not decrypt enough of the shares to reconstruct m .

Now suppose that we give person id the secret key for every attribute of his biometric ω . If someone would send him a message, they would use a noisy biometric ω' , with $d(\omega, \omega') \leq t$. This means that he does have enough of the secret keys to reconstruct m . Thus we have the basics of a simple fuzzy IBE scheme. See Figure 6.1.

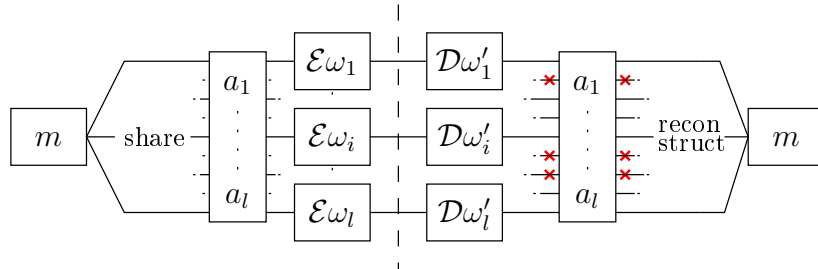


Figure 6.1: Message m is shared. Each share gets encrypted. If enough shares are correctly decrypted, m can be recovered.

However, this scheme would be extremely sensitive to collusion attacks. An adversary could just piece together different secret keys from different identities to create the secret key he requires, or even shuffle positions within one secret key. So we must create some sort of interdependence between the element of the secret key.

Our solution is as follows: we use the combination (v, ω) as a public key. The i -th share will be **BasicIdent**-encrypted with the concatenation $v||i||\omega_i$; the associated secret key will be the collection of **BasicIdent** secret keys associated with $v||i||\omega_i$. Thus in the **BasicIdent** case the secret key associated to (v, ω) would be

$$S = \langle S_1, \dots, S_l \rangle \quad \text{with} \quad S_i = H_2(v||i||\omega_i)^s$$

The next step will be to make sure that message m actually can be reconstructed. The **BasicIdent** scheme gives no information on correct decryption, but outputs a random message

when an incorrect key has been used. When the recipient has received l **BasicIdent** encryptions and has enough secret keys, he needs some way to find out which shares are decrypted correctly in order to reconstruct m .

This behaviour can be created by adding some extra redundancy to the encryptions. Instead of just the share, we encrypt the share together with a cyclic redundancy check. After decryption the receiver can recompute this check to see whether the decryption succeeded.

This leads to the scheme in Figure 6.1. The identity/biometric pair (v, ω) form the public key. s is a master key, and $S = (S_1, \dots, S_l)$, $S_i = H_2(v \| i \| \omega_i)$ forms the secret key.

Encryption is done by sharing message m into l shares, then use **BasicIdent** to encrypt the i -th share together with a cyclic redundancy check using $v \| i \| \omega_i$ as a secret key. To shorten the ciphertext the same nonce can be used in every encryption and sent only once.

The receiver uses the l secret keys S_1, \dots, S_l to decrypt the i -th share and computes the redundancy check to see which shares are decrypted correct. If at least $l - t$ of the shares are correct he can reconstruct m from them.

This way we reach the desired functionality: if two biometric measurements are from the same person then at least $l - t$ of the attributes will be the same. In this case an encryption with one biometric measurement can be decrypted with the secret key derived from another, since enough of the shares can be decrypted to reconstruct m .

On the other hand, if two biometric measurements are from different persons they will differ in at least t positions. In this case decryption will fail, because the receiver will not be able to decrypt enough shares to reconstruct m .

Note. Obviously we need to review the security model for identity-based encryption that we gave in Section 3.2. In the ID model, we give an adversary access to a *private key extraction oracle*. The adversary is allowed to (adaptively) request the private key for any identity, as long as this key cannot directly decrypt the challenge.

In a regular IBE setting, this means that the adversary is not allowed to request the secret key for the identity it wants to be challenged upon. In the fuzzy IBE setting, this restriction extends to secret keys for biometrics that are within the threshold of the biometric to be challenged upon.

Though this falls within our definition of the ID model, it differs from the definition used in other papers. To emphasize the difference, we will use the notation FID in the security proofs.

6.3 Chosen Plaintext Indistinguishability

The following lemma states that scheme **BasicFIBE** is IND-FID-CPA secure, by reducing security to that of the **BasicIdent** scheme.

Lemma 6.1. *Let \mathcal{A} be an IND-FID-CPA adversary that has advantage $\epsilon(\kappa)$ against scheme **BasicFIBE** in Figure 6.1. Suppose \mathcal{A} makes at most $q_E > 0$ private key extraction queries. Then there is a IND-ID-CPA adversary \mathcal{B} that has advantage $\frac{\epsilon(\kappa)}{2}$ against **BasicIdent** in Figure 4.1, while making at most $(l + 1)q_E - 1$ private key extraction queries. Its running time is $O(\text{time}(\mathcal{A}))$.*

Proof. We show how to construct a IND-ID-CPA adversary \mathcal{B} that uses \mathcal{A} to gain advantage $\frac{\epsilon(\kappa)}{2}$. The game between the challenger and the adversary \mathcal{B} starts by the challenger running

Setup $\mathcal{G}(\kappa)$: Using security parameter κ , generate BDH parameters to obtain $\mathbb{G}_1, \mathbb{G}_T$ of order q and a pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_T$. Choose a generator $g \in \mathbb{G}_1$, pick a random $s \in \mathbb{Z}_q^*$ and let $g_{pub} = g^s$. Pick cryptographic hash functions $H_1 : \mathbb{G}_T \mapsto \{0, 1\}^n$ and $H_2 : \{0, 1\}^* \mapsto \mathbb{G}_1^*$ for some n , which will be viewed as random oracles in the security proof. Pick a checksum algorithm $CRC : \{0, 1\}^k \mapsto \{0, 1\}^{n-k}$.

The message space is $\mathcal{M} = \{0, 1\}^k$, the cipher text space is $\mathcal{C} = \mathbb{G}_1 \times \{0, 1\}^{n \times l}$. The master-key is s and the published public parameters are

$$\langle l, t, q, \mathbb{G}_1, \mathbb{G}_T, e, n, g, g_{pub}, H_1, H_2, CRC \rangle.$$

Extract $\mathcal{K}(v, \omega)$: Given identity v and biometric ω compute $Q_i = H_2(v \| i \| \omega_i)$ and $S_i = Q_i^s$ for $i = 1 \dots l$. The private key is $S = \langle S_1, \dots, S_l \rangle$.

Encryption $\mathcal{E}(m, v, \omega)$: To encrypt a message $m \in \mathcal{M}$, pick random $r \in \mathbb{Z}_q^*$. Choose a random polynomial $r(x)$ of degree less than $l - t$ satisfying $r(0) = m$.

Create shares a_i as $a_i = r(i) \| CRC(r(i))$ and then compute $Q_i = H_2(v \| i \| \omega_i)$ and $C_i = a_i \oplus H_1(e(Q_i, g_{pub})^r)$ for $i = 1 \dots l$. The ciphertext is

$$C = \langle g^r, C_1, \dots, C_l \rangle.$$

Decryption $\mathcal{D}(C, S)$: Let $C = \langle U, V_1, \dots, V_l \rangle$ be a ciphertext created using the public key $\langle v, \omega' \rangle$. To decrypt C using private key S compute

$$a_i = V_i \oplus H_1(e(S_i, U)) \quad \text{for } i = 1 \dots l.$$

Let m_i be the first k bits of r_i and let L be the set of all i for which a_i is of the form $m_i \| crc(m_i)$. Find the unique polynomial $r(x)$ of degree less than $l - t$ passing through the points (i, m_i) for $i \in L$. Finally, let $m = r(0)$.

Figure 6.1: Basic fuzzy identity based encryption scheme BasicFIBE

the setup-algorithm $\mathcal{G}(\kappa)$. The result are the public parameters $\langle q, \mathbb{G}_1, \mathbb{G}_2, e, n, g, g_{pub}, H_1, H_2 \rangle$. The challenger gives these parameters to algorithm \mathcal{B} . \mathcal{B} is supposed to choose an identity id and output two messages m_0 and m_1 and expects to receive back the **BasicIdent** encryption of m_b under identity id , for random $b \in \{0, 1\}$. Then algorithm \mathcal{B} outputs its guess $b' \in \{0, 1\}$ for b . Algorithm \mathcal{B} works by interacting with \mathcal{A} in an IND-FID-CPA game as follows (\mathcal{B} simulates the challenger for \mathcal{A}):

Algorithm \mathcal{B} gives \mathcal{A} the public BasicFIBE parameters $\langle l, t, q, \mathbb{G}_1, \mathbb{G}_2, e, n, g, g_{pub}, H_1, H_2 \rangle$. Note that \mathcal{B} does not know the masterkey s that corresponds to these parameters.

At any time \mathcal{A} is allowed to make private key extraction queries. On a query for identity (v, ω) \mathcal{B} responds by doing for $i = 1 \dots l$ a **BasicIdent** private key extraction $S_i = \mathcal{K}(v \| i \| \omega_i)$. She then returns $S = \langle S_1, \dots, S_l \rangle$.

Eventually \mathcal{A} will output an identity (v, ω) and two messages m_0 and m_1 . \mathcal{B} then sends m_0 and m_1 to the challenger together with identity $id = v \| 1 \| \omega_1$. The challenger flips a coin $b \in \{0, 1\}$ and responds with the challenge $C = \langle U, V \rangle$, an encryption of m_b . \mathcal{B} flips a random

coin $c \in \{0, 1\}$ and uses message m_c to create the shares a_i as in the encryption algorithm \mathcal{E} . \mathcal{B} sets $V_1 = V \oplus m_c \oplus a_1$. Then for $i = 2, \dots, l$ she does a private key extraction $S_i = \mathcal{K}(v\|i\|\omega_i)$ and sets $V_i = a_i \oplus H_2(e(S_i, U))$. She hands the ciphertext $C' = \langle U, V_1, V_2, \dots, V_l \rangle$ to \mathcal{A} .

Finally, \mathcal{A} will output a guess c' for c . Then \mathcal{B} will output guess $b' = c'$ for b .

We now proceed to analyze algorithm \mathcal{B} . First we note that the value U is uniformly distributed in \mathbb{G}_1 and because the a_i are random, V_i is uniformly distributed in $\{0, 1\}^n$. Thus C' has the proper distribution for a **BasicFIBE** encryption.

Suppose the challenger flips his coin as b . Then the ciphertext C is

$$C = \langle U, V \rangle = \langle P^r, m_b \oplus H_1(e(H_2(v\|1\|\omega_1), g_{pub})^r) \rangle.$$

If \mathcal{B} flips her coin as c then

$$\begin{aligned} V_1 &= V \oplus m_c \oplus a_1 \\ &= m_b \oplus H_1(e(H_2(v\|1\|\omega_1), g_{pub})^r) \oplus m_c \oplus a_1 \end{aligned}$$

$$\begin{aligned} (\text{if } b = c) \\ &= a_1 \oplus H_1(e(H_2(v\|1\|\omega_1), g_{pub})^r). \end{aligned}$$

For $i = 2, \dots, l$,

$$\begin{aligned} V_i &= a_i \oplus H_1(e(S_i, U)) \\ &= a_i \oplus H_1(e(H_2(v\|i\|\omega_i)^s, g^r)) \\ &= a_i \oplus H_1(e(H_2(v\|i\|\omega_i), g^s)^r) \\ &= a_i \oplus H_1(e(H_2(v\|i\|\omega_i), g_{pub})^r) \end{aligned}$$

Thus, when $b = c$, the ciphertext C' is a valid ciphertext of message m_c with identity (v, ω) and randomness r . \mathcal{A} has an advantage of $\epsilon(\kappa)$ of making the right guess for c . If $b \neq c$, then C' is not a valid ciphertext (though it is in the correct distribution) and \mathcal{A} does not have any advantage.

The chance $Pr[b = c] = \frac{1}{2}$. Thus the chance that \mathcal{B} does a correct guess is

$$\begin{aligned} Pr[b = b'] &= Pr[b = c'] \\ &= Pr[b = c]Pr[c = c'] + Pr[b \neq c]Pr[c' = c'] \\ &= \frac{1}{2} \cdot \left(\frac{1}{2} + \epsilon(\kappa)\right) + \frac{1}{2} \cdot \frac{1}{2} \\ &= \frac{1}{2} + \frac{\epsilon(\kappa)}{2} \end{aligned}$$

Thus \mathcal{B} has an advantage of $\frac{\epsilon(\kappa)}{2}$. Finally, for every extraction query that \mathcal{A} does, \mathcal{B} must do l

extraction queries. During the construction of the ciphertext, \mathcal{B} does another $l - 1$ extraction queries. So if \mathcal{A} does at most q_E extraction queries, then \mathcal{B} does at most $(l + 1)q_E - 1$ private key extraction queries. \square

When we combine Lemma 6.1 with the IND-ID-CPA security of the **BasicIdent** scheme by Boneh and Franklin we can conclude that the **BasicFIBE** scheme is IND-FID-CPA secure:

Theorem 6.2. *Suppose that the hash functions H_1 and H_2 are random oracles. Then the **BasicFIBE** scheme in Figure 6.1 is IND-FID-CPA secure assuming the BDH problem is hard relative to the generated BDH parameters.*

*Concretely, suppose there is an IND-FID-CPA adversary \mathcal{A} that has advantage $\epsilon(\kappa)$ against the scheme **BasicFIBE**. Suppose \mathcal{A} makes at most $q_E > 0$ private key extraction queries and $q_{H_2} > 0$ hash queries to H_2 . Then there is an algorithm \mathcal{B} that solves the BDH in the generated groups with advantage at least:*

$$\text{Adv}(\kappa) \geq \frac{\epsilon(\kappa)}{e(l+1)q_E(q_{H_2} + l - 1)},$$

Here e is the base of the natural logarithm. The running time of \mathcal{B} is $O(\text{time}(\mathcal{A}))$.

Proof. This follows directly by combining Theorem 4.1 and Lemma 6.1. \square

6.4 Chosen Plaintext Anonymity

The following lemma states that scheme **BasicFIBE** is ANO-FID-CPA secure, by reducing security to that of the **BasicIdent** scheme.

Lemma 6.3. *Let \mathcal{A} be an ANO-FID-CPA adversary that has an advantage $\epsilon(\kappa)$ against **BasicFIBE** in Figure 6.1. Suppose \mathcal{A} makes at most q_E private key extraction queries. Then there is a ANO-ID-CPA adversary \mathcal{B} that has advantage $\frac{\epsilon(\kappa)}{2}$ against scheme **BasicIdent** in Figure 4.1, while making at most $(l+1)q_E - 1$ private key extraction queries. Its running time is $O(\text{time}(\mathcal{A}))$.*

Proof. We show how to construct a ANO-ID-CPA adversary \mathcal{B} that uses \mathcal{A} to gain advantage $\frac{\epsilon(\kappa)}{2}$. The game between the challenger and the adversary \mathcal{B} starts by the challenger running the setup-algorithm $\mathcal{G}(\kappa)$, resulting in the public parameters $\langle q, \mathbb{G}_1, \mathbb{G}_2, e, n, g, g_{\text{pub}}, H_1, H_2 \rangle$. The challenger gives these parameters to algorithm \mathcal{B} . \mathcal{B} is supposed to choose a message m and two identities id_0 and id_1 and expects to receive back the **BasicIdent** encryption of m under identity id_b , for random $b \in \{0, 1\}$. Then algorithm \mathcal{B} outputs its guess $b' \in \{0, 1\}$ for b . Algorithm \mathcal{B} works by interacting with \mathcal{A} in an ANO-FID-CPA game as follows (\mathcal{B} simulates the challenger for \mathcal{A}):

Algorithm \mathcal{B} gives \mathcal{A} the public **BasicFIBE** parameters $\langle l, t, q, \mathbb{G}_1, e, n, g, g_{\text{pub}}, H_1, H_2 \rangle$. Note that \mathcal{B} does not know the masterkey s that corresponds to these parameters.

At any time \mathcal{A} is allowed to make private key extraction queries. On a query for identity (v, ω) \mathcal{B} responds by doing for $i = 1 \dots l$ a **BasicIdent** private key extraction $S_i = \mathcal{K}(v \| i \| \omega_i)$. She then returns $S = \langle S_1, \dots, S_l \rangle$.

Eventually, \mathcal{A} will output a message m and two identities (v_0, ω_0) and (v_1, ω_1) . \mathcal{B} then sends m to the challenger together with identities $id_0 = v \| 1 \| \omega_1$ and $id_1 = v' \| 1 \| \omega'_1$. The challenger flips a coin $b \in \{0, 1\}$ and responds with the challenge $C = \langle U, V \rangle$, an encryption of m under identity id_b .

\mathcal{B} uses message m to create the shares a_i as in the encryption algorithm \mathcal{E} . She sets $V_1 = V \oplus m \oplus a_1$. \mathcal{B} then flips a random coin $c \in \{0, 1\}$, she does a private key extraction $S_i = \mathcal{K}(v_c \| i \| \omega_{ci})$ and sets $V_i = a_i \oplus H_2(e(S_i, U))$ for $i = 2, \dots, l$. She hands the ciphertext $C' = \langle U, V_1, V_2, \dots, V_l \rangle$ to \mathcal{A} .

Finally, \mathcal{A} will output a guess c' for c . Then \mathcal{B} will output guess $b' = c'$ for b .

Now we proceed to analyze algorithm \mathcal{B} . First we note that the value U is uniformly distributed in \mathbb{G}_1 and because the a_i are random, V_i is uniformly distributed in $\{0, 1\}^n$. Thus C' has the proper distribution for a **BasicFIBE** encryption. Suppose the challenger flips his coin as b . Then the ciphertext C is

$$C = \langle U, V \rangle = \langle g^r, m \oplus H_1(e(H_2(v_b \| 1 \| \omega_{b_1}), g_{pub})^r) \rangle$$

and thus

$$\begin{aligned} V_1 &= V \oplus m \oplus a_1 \\ &= m \oplus H_1(e(H_2(v_b \| 1 \| \omega_{b_1}), g_{pub})^r) \oplus m \oplus a_1 \\ &= a_1 \oplus H_1(e(H_2(v_b \| 1 \| \omega_{b_1}), g_{pub})^r) \end{aligned}$$

If \mathcal{B} flips her coin as c then for $i = 2, \dots, l$

$$\begin{aligned} V_i &= a_i \oplus H_2(e(S_i, U)) \\ &= a_i \oplus H_2(e(H_2(v_c \| i \| \omega_{c_i})^s, g^r)) \\ &= a_i \oplus H_2(e(H_2(v_c \| i \| \omega_{c_i}), g^s)^r) \\ &= a_i \oplus H_2(e(H_2(v_c \| i \| \omega_{c_i}), g_{pub})^r) \end{aligned}$$

Thus, when $b = c$, the ciphertext C' is a valid ciphertext of message m with identity (v_c, ω_c) and randomness r . \mathcal{A} has an advantage of $\epsilon(\kappa)$ in making the right guess for c . If $b \neq c$, then C' is not a valid ciphertext (though it is in the correct distribution) and \mathcal{A} does not have any advantage.

The chance that \mathcal{B} does a correct guess is

$$\begin{aligned} Pr[b = b'] &= Pr[b = c'] \\ &= Pr[b = c]Pr[c = c'] + Pr[b \neq c]Pr[c \neq c'] \\ &= \frac{1}{2} \cdot \left(\frac{1}{2} + \epsilon(\kappa)\right) + \frac{1}{2} \cdot \frac{1}{2} \\ &= \frac{1}{2} + \frac{\epsilon\kappa}{2} \end{aligned}$$

and thus \mathcal{B} has an advantage of $\frac{\epsilon\kappa}{2}$.

Finally, for every extraction query that \mathcal{A} does, \mathcal{B} must do l extraction queries. During the construction of the ciphertext, \mathcal{B} does another $l - 1$ extraction queries. So if \mathcal{A} does at most q_E extraction queries, then \mathcal{B} does at most $(l + 1)q_E - 1$ private key extraction queries. \square

6.5 Chosen Ciphertext Attacks

We can now use the Fujisaki-Okamoto transformation to strengthen **BasicFIBE** to security under chosen ciphertext attacks. We have already shown that this scheme has anonymity and indistinguishability. Theorem 3.8 then states that it is also CI-FID-CPA secure.

The last thing we required before we can apply Fujisaki-Okamoto is the γ -uniformity, as defined in 4.3:

Lemma 6.4. *The BasicFIBE scheme from Figure 6.1 is $((q-1)2^{ln})^{-1}$ -uniform.*

Proof. Let Π be a BasicFIBE scheme with master-secret s . Then for any public key (v, ω) , message x and ciphertext $y = \langle U, V_1, \dots, V_l \rangle$

$$\begin{aligned}
\gamma_{(v, \omega)}(x, y) &= \Pr[\exists r : y = \mathcal{E}_{(v, \omega)}(x; r)] \\
&= \Pr[\exists r : U = g^r \wedge V_1 = a_1 \oplus H_1(e(H_2(v\|1\|\omega_1), g^s)^r) \wedge \dots \wedge V_l = a_l \oplus H_1(e(H_2(v\|l\|\omega_l), g^s)^r)] \\
&= \Pr[\exists r : U = g^r] \prod_{i=1}^l \Pr[V_i = a_i \oplus H_1(e(H_2(v\|i\|\omega_i), g^s)^r) \mid U = g^r] \\
&= \Pr[\exists r : U = g^r] \prod_{i=1}^l \Pr[H_1(e(H_2(v\|i\|\omega_i), U^s)) = V_i \oplus a_i] \\
&\leq \frac{1}{q-1} \prod_{i=1}^l 2^{-n} \\
&= \frac{1}{q-1} 2^{-ln}
\end{aligned}$$

Thus Π is $((q-1)2^{ln})^{-1}$ -uniform. \square

This leads to scheme FullFIBE in Figure 6.2.

Theorem 6.5. *Scheme FullFIBE in Figure 6.2 is CI-FID-CCA secure.*

Proof. Combining Lemma's 6.1 and 6.3, and Theorem 3.8, gives as a result that BasicFIBE is CI-FID-CPA secure. According to Lemma 5.2, it also secure in the sense of Game-1.

Then Theorem 5.5 says that it is also CI-FID-CCA secure. \square

Setup $\mathcal{G}(\kappa)$: Using security parameter κ , generate BDH parameters to obtain $\mathbb{G}_1, \mathbb{G}_T$ of order q and a pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_T$. Choose a generator $g \in \mathbb{G}_1$, pick a random $s \in \mathbb{Z}_q^*$ and let $g_{pub} = g^s$. Pick cryptographic hash functions $H_1 : \mathbb{G}_T \mapsto \{0, 1\}^n$, $H_2 : \{0, 1\}^* \mapsto \mathbb{G}_1^*$ for some n , $H_3 : \{0, 1\}^* \mapsto \mathbb{Z}_q^*$ and $H_4 : \{0, 1\}^* \mapsto K$. Pick a checksum algorithm $CRC : \{0, 1\}^k \mapsto \{0, 1\}^{n-k}$. Let $E^{sym} : \mathcal{M} \times K \mapsto \{0, 1\}^*$ be an IND-SYM secure symmetric cipher,

The message space is $\mathcal{M} = \{0, 1\}^n$, the cipher text space is $\mathcal{C} = \mathbb{G}_1 \times \{0, 1\}^{n \times l}$. The master-key is s and the published public parameters are

$$\langle l, t, q, \mathbb{G}_1, \mathbb{G}_T, e, n, g, g_{pub}, H_1, H_2, CRC \rangle.$$

Extract $\mathcal{K}(v, \omega)$: Given identity v and biometric ω compute $Q_i = H_2(v \| i \| \omega_i)$ and $S_i = Q_i^s$ for $i = 1 \dots l$. The private key is $S = \langle S_1, \dots, S_l \rangle$.

Encryption $\mathcal{E}(m, v, \omega)$: To encrypt a message $m \in \mathcal{M}$, pick random $\sigma \in \{0, 1\}^k$. Choose a random polynomial $r(x)$ of degree less than $l - t$ satisfying $r(0) = \sigma$. Compute $r = H_3(m, \sigma)$.

Create shares a_i as $a_i = r(i) \| CRC(r(i))$ and then compute $Q_i = H_2(v \| i \| \omega_i)$ and $C_i = a_i \oplus H_1(e(Q_i, g_{pub})^r)$ for $i = 1 \dots l$.

The ciphertext is

$$C = \langle g^r, C_1, \dots, C_l, E_{H_4(\sigma)}^{sym}(m) \rangle.$$

Decryption $\mathcal{D}(C, S)$: Let $C = \langle U, V_1, \dots, V_l, W \rangle$ be a ciphertext created using the public key $\langle v, \omega' \rangle$. To decrypt C using private key S compute

$$a_i = V_i \oplus H_1(e(S_i, U)) \quad \text{for } i = 1 \dots l.$$

Let σ_i be the first k bits of r_i and let L be the set of all i for which a_i is of the form $\sigma_i \| crc(\sigma_i)$. Find the unique polynomial $r(x)$ of degree less than $l - t$ passing through the points (i, m_i) for $i \in L$ and let $\sigma = r(0)$. Decrypt $m = D_{H_4(\sigma)}^{sym}(W)$ and verify that $U = g^{H_3(m, \sigma)}$.

Figure 6.2: Fuzzy identity based encryption scheme FullFIBE

Chapter 7

Conclusions

In this thesis we constructed a new identity-based encryption scheme, which allows for error-tolerance between the identity of the private key and the public key used to encrypt the ciphertext. The construction is presented using Hamming distance as a distance metric between identities. We gave full proof for security in both the sense of indistinguishability and anonymity, both in a chosen ciphertext attack model.

Performance of the scheme is very poor, since the amount of pairings that must be computed is linear in the length of the identity's biometric. Since pairings are an expensive operation and biometric often need long descriptions, the amount of computations can be enormous. Both encryption and decryption is dominated by l pairings.

7.1 Comparison with Sahai-Waters

It is interesting to compare our construction with the construction of [SW05], which is the only other work on Fuzzy IBE. We can find the following clear distinctions:

- First of all our scheme has the anonymity property, thus it hides the public key used for encryption. Sahai-Waters send the public key unencrypted with the ciphertext.
- Sahai-Waters strictly uses a biometric as public key, where our scheme uses a biometric in combination with an identity.
- Our scheme is proven secure in adaptive-ID, where Sahai-Waters is proven in the weaker selective-ID. As a trade-off we use the random oracle model.
- In our scheme both encryption and decryption are dominated by l pairing operations. In Sahai-Waters encryption is dominated by l exponentiations and decryption by $2l$ pairing operations.
- Sahai-Waters security is based on the Decisional Bilinear Diffie-Hellman (DBDH) assumption; Our scheme is based on Boneh-Franklin and indirectly on the weaker Bilinear Diffie-Hellman assumption. [CL02]
- Both schemes use Fujisaki-Okamoto [FO99] to reach security under chosen ciphertext attacks, though Sahai-Waters only mentions the possibility. The reduction for this transformation is not tight, which is a problem for any scheme using this technique.

7.2 Further Research

The field of Fuzzy Identity-Based Encryption is a new terrain with lots of room for improvements and new research. We have some ideas for improvement for which there was no time and also some interesting open problems.

With the use of some relatively new techniques, it is possible to improve the encryption scheme we presented on several fronts:

- We use the transformation from Fujisaki-Okamoto [FO99] to reach security under chosen ciphertext attacks. This transform is very efficient, but the security proof is not tight and based on random oracles.

The security proof could be tightened by using an approach from Boneh-Katz [BK05]. Their method can probably be adapted to this setting. It would result in a tight reduction in the standard model, at the cost of computational complexity. However, these extra computations will be negligible to the cost of the pairings used for BasicFIBE.

- We used Boneh-Franklin's BasicIdent [BF01] as a basis for our BasicFIBE scheme. Instead of that, we can substitute any other IBE scheme that is both IND-FID-CPA and ANO-FID-CPA secure.

In particular, Boyen and Waters describe in [BW06] a IBE scheme that is anonymous under selective-id assumptions. This scheme can be transformed to an adaptive-id setting by using a hashing technique due to Waters [Wat05].

Using our approach, the resulting scheme can be transformed into a CI-FID-CPA secure scheme that is secure in the standard model. Again, this goes at the cost of additional computational complexity.

- As mentioned in Chapter 6, our approach at creating a fuzzy IBE is inherently vulnerable to collusion attacks. This attack is possible by combining several requested secret keys. To resolve this problem we had to include a fixed identity v in the public key, which limits its applications.

A better solution would be to make the secret keys probabilistic, i.e. multiple requests for a secret key for the same identity results in completely different secret keys. This should be possible with ideas lined out in [Wat05].

With these improvements it would be possible to reach every application given in the introduction, among which would be Public-key Encryption with Fuzzy keyword search. We used these ideas to construct the alternative scheme in Appendix A. This is purely for demonstration of the ideas however, so no proofs are included.

We also have a few open problems that may be interesting for further research.

- What would happen if you give adversaries access to an ‘identification oracle’: An oracle that takes a ciphertext and returns the public key or identity it was encrypted with.

This will model the scenario in which an adversary knows the recipient of a few ciphertexts. Is it then possible to learn anything about other encryptions. Will the current anonymous IBE schemes still be anonymous?

- An interesting scenario would be a Fuzzy Identity-based encryption scheme where the attributes come from multiple authorities. This is especially useful in attribute-based encryption applications.

Since the basis of our scheme is a set of independent, parallel IBE schemes this should not be very hard to accomplish.

- Our Fuzzy IBE scheme is based on the Hamming distance combined with a threshold secret sharing scheme, so a secret can be recovered if there are enough matches.

With an eye on attribute-based encryption and encryption with keyword search, it is interesting whether other access structures can be implemented. Also interesting is the point in the scheme where this access structure is chosen: during the setup, during secret key generation, or during encryption. Each of these possibilities leads to completely different applications.

Bibliography

- [ABC⁺05] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 205–222. Springer, 2005.
- [APM04] Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors. *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, October 25-29, 2004*. ACM, 2004.
- [BB04] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459. Springer, 2004.
- [BBDP01] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT*, pages 566–582, 2001.
- [BCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Cachin and Camenisch [CC04], pages 506–522.
- [BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO*, pages 26–45, 1998.
- [BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.
- [BK05] Dan Boneh and Jonathan Katz. Improved efficiency for cca-secure cryptosystems built using identity-based encryption. In Alfred Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 87–103. Springer, 2005.
- [BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
- [BR94] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In *EUROCRYPT*, pages 92–111, 1994.

- [BW06] Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307. Springer, 2006.
- [CC04] Christian Cachin and Jan Camenisch, editors. *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*. Springer, 2004.
- [CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.
- [CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271. Springer, 2003.
- [CL02] J. Cheon and D. Lee. Diffie-hellman problems and bilinear maps, 2002.
- [Cra05] Ronald Cramer, editor. *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*. Springer, 2005.
- [DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.
- [DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Cachin and Camenisch [CC04], pages 523–540.
- [FMR99] Gerhard Frey, Michael Müller, and Hans-Georg Rück. The tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *IEEE Transactions on Information Theory*, 45(5):1717–1719, 1999.
- [FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. pages 537–554, 1999.
- [GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.
- [GPS06] S.D. Galbraith, K.G. Paterson, and N.P. Smart. Pairings for cryptographers. Cryptology ePrint Archive, Report 2006/165, 2006. <http://eprint.iacr.org/>.
- [Jou00] Antoine Joux. A one round protocol for tripartite diffie-hellman. In Wieb Bosma, editor, *ANTS*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394. Springer, 2000.
- [Jou02] Antoine Joux. The weil and tate pairings as building blocks for public key cryptosystems. In Claus Fieker and David R. Kohel, editors, *ANTS*, volume 2369 of *Lecture Notes in Computer Science*, pages 20–32. Springer, 2002.
- [Mil86] V. Miller. Short program for functions on curves, 1986.

-
- [MOV93] Alfred Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.
- [Nao98] Moni Naor. Private communication. 1998.
- [Nie02] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 111–126. Springer, 2002.
- [NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437, 1990.
- [RS91] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO*, pages 433–444, 1991.
- [Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.
- [SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Cramer [[Cra05](#)], pages 457–473.
- [SWP00] Dawn Xiaodong Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *IEEE Symposium on Security and Privacy*, pages 44–55, 2000.
- [Tsu92] Gene Tsudik. Message authentication with one-way hash functions. In *INFOCOM*, pages 2055–2059, 1992.
- [Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In Cramer [[Cra05](#)], pages 114–127.

Appendix A

Alternative Fuzzy Scheme

In Section 7.2, we gave some possible improvements to create a more general applicable Fuzzy IBE. We used a few of the ideas to create the following encryption schemes.

This scheme is without proof, as we did not have the time to complete it. We suspect it is possible to prove CI-FID-CCA security, based on other papers. The proofs would be completely different from the ones in this thesis however.

We will show the correctness of decryption. Let ciphertext C be an encryption under biometric ω' , which we will decrypt with secret key S based on ω . Then if and only if $w_i = w'_i$

$$\begin{aligned} e(c_{1i}, H(i||\omega_i)) &= e(g^{r_i}, H(i||\omega_i)) \\ &= e(g^{r_i}, H(i||\omega'_i)) \\ &= e(g, H(i||\omega'_i)^{r_i}) \\ &= e(c_{2i}, g), \end{aligned}$$

thus $D = \{i \in \{1 \dots l\} \mid \omega_i = \omega'_i\}$. For every $i \in D$ the following holds:

$$\begin{aligned} \frac{e(c_{1i}, S_i)}{e(c_{2i}, S_0)} &= \frac{e(g^{r_i}, \alpha \cdot Q_i^\rho)}{e(Q_i^{r_i}, g^\rho)} \\ &= \frac{e(g^{r_i}, Q_i^\rho)}{e(Q_i^{r_i}, g^\rho)} \cdot e(g^{r_i}, \alpha) \\ &= \frac{e(g, Q_i)^{\rho r_i}}{e(Q_i, g)^{\rho r_i}} \cdot e(g, \alpha)^{r_i} \\ &= e(g, \alpha)^{r_i} \\ &= z^{r_i}. \end{aligned}$$

Thus

$$\begin{aligned}
K &= \prod_{i \in D'} \left(\frac{e(c_{1i}, S_i)}{e(c_{2i}, S_0)} \right)^{-\prod_{j \in D' \setminus \{i\}} \frac{i}{i-j}} \\
&= \prod_{i \in D'} (z^{r_i})^{-\prod_{j \in D' \setminus \{i\}} \frac{i}{i-j}} \\
&= z^{\sum_{i \in D'} (-r_i \prod_{j \in D' \setminus \{i\}} \frac{i}{i-j})} \\
&= z^{\sum_{i \in D'} (-r(i) \prod_{j \in D' \setminus \{i\}} \frac{i}{i-j})} \\
&= z^r,
\end{aligned}$$

where the last line is derived from using polynomial interpolation in the exponents. Since the polynomial $r(x)$ is of degree $l - t - 1$ it can be interpolated using $l - t$ points.

Finally, symmetric decryption is done with key $G(K)$. This is the same key as used during encryption, which results in the original message m .

The encryption is dominated by $2l$ exponentiations. Decryption is dominated by approximately $4l$ pairing operations.

Setup $\mathcal{G}(\kappa)$: Using security parameter κ , generate BDH parameters to obtain \mathbb{G}_1 , \mathbb{G}_T of order q and a pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_T$. Choose a generator $g \in \mathbb{G}_1$, pick a random $s \in \mathbb{Z}_q^*$, let $\alpha = g^s$ and $z = e(g, \alpha)$. Pick a symmetric cipher E^{sym} with keyspace \mathcal{K} . Pick cryptographic hash functions $G : \mathbb{G}_T \mapsto \mathcal{K}$ and $H : \{0, 1\}^* \mapsto \mathbb{G}_1^*$ for some n ,

The message space is $\mathcal{M} = \{0, 1\}^n$, the cipher text space is $\mathcal{C} = \mathbb{G}_1 \times \{0, 1\}^{n \times l}$. The master-key is α and the published public parameters are

$$\langle l, t, q, \mathbb{G}_1, \mathbb{G}_T, e, n, g, z, G, H \rangle.$$

Extract $\mathcal{K}(\omega)$: Given biometric $\omega = (\omega_1, \dots, \omega_l)$ compute $Q_i = H(i \parallel \omega_i)$. Pick a random ρ , let $S_0 = g^\rho$ and $S_i = \alpha \cdot Q_i^\rho$ for $i = 1 \dots l$. The private key is $S = \langle S_0, S_1, \dots, S_l \rangle$.

Encryption $\mathcal{E}(m, \omega)$: To encrypt a message $m \in \mathcal{M}$, pick random $r \in \mathbb{Z}_q^*$. Choose a random polynomial $r(x)$ of degree less than $l - t + 1$ satisfying $r(0) = r$. For $i = 1 \dots l$ compute $r_i = r(i)$, $c_{1i} = g^{r_i}$, $Q_i = H(i \parallel \omega_i)$ and $c_{2i} = Q_i^{r_i}$.

Let $c_0 = E_{G(z^r)}^{sym}(m)$ and create the ciphertext

$$C = \langle c_0, (c_{1i}, c_{2i})_{i=1}^l \rangle.$$

Decryption $\mathcal{D}(C, S)$: Let $C = \langle c_0, (c_{1i}, c_{2i})_{i=1}^l \rangle$ be a ciphertext created using the public key ω' . To decrypt C using private key S , which is derived from public key ω do the following:

Let

$$D = \{i \in \{1 \dots l\} \mid e(c_{1i}, H(i \parallel \omega_i)) = e(c_{2i}, g)\}$$

. Choose an arbitrary $l - t$ -element subset $D' \subset D$, and compute

$$K = \prod_{i \in D'} \left(\frac{e(c_{1i}, S_i)}{e(c_{2i}, S_0)} \right)^{-\prod_{j \in D' \setminus \{i\}} \frac{i}{i-j}}.$$

Now $m = D_{G(K)}^{sym}(c_0)$.

Figure A.1: Alternative fuzzy identity based encryption scheme